מבוא :

מערכות זיהוי דיבור כקלט להפעלת מחשב נמצאות בשימוש הציבור מאז סוף שנות ה-80. נכון לסוף העשור הראשון של המאה ה-21 אדם בעל תפקוד ידיים תקין הנמצא במצב שבו ידיו פנויות להקלדה, יחבר תמליל מהר יותר בהקלדה מאשר בדיבור אל המחשב. זאת משום שתיקון שגיאות הקלט הרבות של מערכות אלה גוזל זמן רב. מקליד זריז עשוי להגיע לקצב הקלדה של 100 מילים בדקה, בעוד שאדם מיומן בחיבור תמליל בעזרת מערכת לזיהוי דיבור יגיע לפחות מ-50 מילים לדקה וגם זאת אחרי שהמחשב כוונן לקולו ואחרי שהשגיאות הטיפוסיות לקלט מהדיבור שלו הוזנו בפונקציית תיקון השגיאות האוטומטית. מעבר לזאת, במשימות הקלדה קצרות (שני משפטים ומטה), נכה שכיוון את מערכת זיהוי הקול שלו על ידי מומחה יכול להשיג תוצאות דומות לאלה של אדם חסר לקות. יש אנשים שההקלדה על מקלדת גרמה להם פציעות מאמץ חוזרני, כגון תסמונת תעלת שורש היד. חלק מהאנשים האלה עוברים לשימוש במערכת זיהוי דיבור במקום במקלדת כלשהי. [1]

המערכת שלנו תתבסס על NLU/NLP – Natural Language Understanding (עיבוד שפה טבעית). זהו למעשה אחד מתחומי הבינה מלאכותית ותפקידו לחקור ולפתח סוגיות הקשורות לעיבוד של שפה טבעית, בשילוב הבנת השפה הטבעית של בני האדם. זאת על מנת לסייע למחשבים (מערכות טכנולוגיות, אפליקציות לזיהוי קולי, אפליקציות לאינטרנט ועוד) "להבין" דברים שנאמרים בשפה טבעית אנושית. היכולות העיקריות של NLU\NLP: זיהוי דיבור, הבנת דיבור, תרגום, הבנת שפה טבעית, חיפוש מידע, בדיקה וסריקה של טקסטים, יצירת תקצירי מלל אוטומטים. [4]

פיתוח אפליקציה לזיהוי דיבור ותמלול היא דבר מתבקש בימינו לאור התפתחות הטכנולוגיות והניסיון להקל על בני האדם ולייעל את זמנם. כיום קיימות אפליקציות רבות לזיהוי קול ותמלול (Siri,Google now, amazon lex, etc) אך מבדיקה מרחיבה שלנו אין בהן שלמות והרעיון שלנו הוא להוסיף להם פיצ'רים נוספים על מנת להתעלות על אותן מערכות קיימות ובנוסף לבצע שמירה אוטומטית של הדיבור שהוקלט והומר לטקסט כקובץ word שישלח למשתמש. בחרנו להתמקד בנושא סימני הפיסוק, במערכת שלנו כל משתמש יוכל לשמור לעצמו מילון אישי ובו מילות מפתח שכל אחת מהן יסמלו סימן פיסוק אחר על מנת שכשהמערכת תקבל כקלט מהמשתמש את המילה :"נקודה" היא תשים : " ." (סימן הפיסוק ולא את המילה עצמה). כמובן שזאת לאחר שהוא יבצע רישום למערכת ויבצע הקלטות לתמלול.

המחשבה על רעיון הפרויקט הגיעה בעקבות שימוש אישי שלנו באחת המערכות והיתקלות בבאגים שעדיין לא נמצא להם פתרון. לכן החלטנו לקחת לידינו את הבעיות ולנסות לפתור אותן ולשדרג מערכות קיימות.

במהלך הפיתוח והמחקר של הפרויקט נדרשנו ללמוד ולחקור עולם תוכן חדש שרובו היה זר לנו. החל מחקר על הנושא בכללי, מה קיים היום בשוק? האם מישהו הצליח להגיע לפתרון שלנו? חשיבה על פיתוח האפליקציה מבחינה רעיונות ותכנותית ? ממשקים קיימים? ובחלק של הפיתוח בחרנו לעבוד עם ערכת הפיתוח Flutter ובשפת Dart. הבחירה נעשתה משיקולי נוחות ומתן גישה לפלטפורמות שונות לשימוש באפליקציה (כידוע flutter הוא multi platform ומשמש לפיתוח אפליקציות עבור אנדרואיד,iOS, Windows, Mac, Linux, Google Fuchsia) [2]. סיבה עיקרית לפופולריות של Dart כשפת תכנות היא העובדה שגוגל בנו בעזרתה את Flutter וכל זאת בצורה נוחה במיוחד [3].

כמהנדסים לעתיד רצינו לבצע פרויקט גמר העוסק בבעיה מהעולם האמיתי, מתוך מחשבה שתפקידו של מהנדס הוא לחפש פתרונות לבעיות יומיומיות ובנייה של תהליך איכותי המבטיח פתרון מיטבי יעיל ובטוח לבעיה זו.

מסמך דרישות

Epic 1 – Entry Page:

1.כמשתמש אוכל ללחוץ על כפתור "התחברות" או "הרשמה" על מנת שאוכל להתחיל לבצע שימוש במערכת.

2.כמפתח ארצה שעמוד הפתיחה של האפליקציה יכיל 2 כפתורים "התחברות" ו-"הרשמה" על מנת שהמשתמש יוכל לבחור מבניהם ולהתחיל להשתמש במערכת.

3. כמפתח ארצה שכאשר המשתמש ילחץ על אחד מן הכפתורים "התחברות" או "הרשמה" הוא יעבור מיד לעמוד המתאים על מנת שיוכל להמשיך את השימוש באפליקציה.

Epic 2 -Login :

1. כמשתמש אוכל להתחבר למערכת פעם אחת ומערכת תזכור אותי עד שאתנתק ממנה, על מנת שלא אצטרך לבצע התחברות בכל שימוש באפליקציה.

2.כמשתמש אוכל לבחור מתוך עמוד ההתחברות במעבר לעמוד ההרשמה על ידי לחיצה על כפתור "הרשמה" על מנת שאוכל לעבור לעמוד ההרשמה במקרה שעדיין אין לי משתמש להתחבר בעזרתו למערכת.

3. כמשתמש אוכל להזין את השם משתמש שלי (אימייל) ואת הסיסמא על מנת שאוכל להתחבר למערכת.

4.כמשתמש אוכל ללחוץ על הכפתור "שכחתי סיסמא" על מנת שאוכל לשחזר את הסיסמא שלי במידה ושכחתי אותה.

5. כמשתמש אוכל ללחוץ על כפתור האישור בעמוד התחברות במידה ושם המשתמש והסיסמא נכונים על מנת לבצע את ההתחברות למערכת.

6. כמפתח ארצה לבדוק אם המשתמש כבר מחובר למערכת כדי שלא יצטרך לבצע התחברות מחדש.

7.כמפתח ארצה שיהיה בעמוד ההתחברות כפתור "הרשמה" שיוביל את המשתמש לעמוד ההרשמה במידה ועדיין אין לו משתמש להתחבר בעזרתו כדי שיוכל להירשם למערכת.

8.כמפתח ארצה שיהיה כפתור "שכחתי סיסמא" על מנת שהמשתמש יוכל לשחזר את הסיסמא שלו במידה ושכח אותה.

9. כמפתח ארצה שתוצג הודעת שגיאה במידה והמשתמש הכניס שם משתמש או סיסמא שלא קיימים ב-database על מנת שהוא ידע לתקן את זה.

10.כמפתח ארצה שתוצג הודעת שגיאה במידה ותיבת הטקסט של השם משתמש או של הסיסמא נשארה ריקה והמשתמש לחץ על אישור , על מנת שידע למלא את התיבה החסרה.

11.כמפתח ארצה שיהיה כפתור של אישור לאחר שהמשתמש סיים למלא את פרטיו על מנת שיוכל להתחבר למערכת.

12.כמפתח ארצה שכאשר המשתמש סיים להכניס את הפרטים שלו ולחץ על אישור , במידה והפרטים נכונים הוא יעבור מיד לעמוד הבית על מנת שיוכל להתחיל את השימוש במערכת.

13.כמפתח ארצה שיהיו 2 תיבות טקסט בעמוד ההתחברות אחת של שם משתמש והשנייה של סיסמא, על מנת שהמשתמש יוכל להזין את פרטיו.

14.כמפתח ארצה שכאשר המשתמש ילחץ על הכפתור "שכחתי סיסמא" הוא יעבור לעמוד נפרד שם יצטרך להזין את כתובת המייל שלו אליה תישלח סיסמא חדשה, על מנת שיוכל לאפס את הסיסמא שלו.

15.כמשתמש אוכל להזין את כתובת המייל שלי בעמוד של "שכחתי סיסמא" על מנת לשחזר את הסיסמא שלי.

Epic 3 – register:

1.כמשתמש אוכל להזין את השם משתמש שלי , סיסמא ואישור סיסמא על מנת שאוכל להירשם למערכת.

2.כמשתמש אוכל ללחוץ על כפתור האישור בעמוד ההרשמה על מנת שאוכל להירשם למערכת ולהתחיל את השימוש בה.

3.כמשתמש אוכל ללחוץ על "אישור תקנון" על מנת להשלים את ההרשמה למערכת.

4.כמפתח ארצה שתוצג הודעת שגיאה כאשר המשתמש הזין שם משתמש לא תקין (מייל לא קיים) על מנת שידע לתקן את זה.

5.כמפתח ארצה שתוצג הודעת שגיאה כאשר המשתמש הזין סיסמא הכוללת פחות מ-6 תווים על מנת שידע לתקן את זה.

6.כמפתח ארצה שתוצג הודעת שגיאה כאשר המשתמש הזין אישור סיסמא ששונה מהסיסמא המקורית על מנת שידע לתקן את זה.

7.כמפתח ארצה שתהיה אופציה לסימון " אישור תקנון" על מנת שהמשתמש יוכל לאשר שהוא מקבל את מדיניות האפליקציה.

8.כמפתח ארצה שתופיע הודעת שגיאה במידה והמשתמש לא סימן אישור על "אישור תקנון" על מנת שהמשתמש לא יוכל להירשם למערכת לפני שקרא ואישר את התנאים.

9.כמפתח ארצה שבמידה והמשתמש מילא את כל תיבות הטקסט כראוי ולחץ על הכפתור אישור הרשמה הוא ירשם למערכת ויעבור לעמוד הבית.

10.כמפתח ארצה שיהיה כפתור לאישור הרשמה בסיום עמוד הרישום על מנת שהמשתמש יוכל ללחוץ עליו ולסיים את הרישום.

11.כמפתח ארצה שתוצג הודעת שגיאה במידה ואחת מתיבות הטקסט (שם משתמש, סיסמא ואישור סיסמא) יישארו ריקות על מנת שהמשתמש ידע למלא אותן.

Epic 4 – Personal Setting:

2. כמשתמש אוכל ללחוץ על כפתור המיקרופון על מנת להתחיל להקליט את הטקסט שלי או את מילות המפתח שלי .

2.כמשתמש אוכל להקליט מילות מפתח שישמשו את המשתמש להגדרת הסימנים שלו כמו ירידת שורה, סימני פיסוק ועוד.. על מנת שהטקסט שלו יהיה מתומלל כראוי.

3.כמפתח ארצה שיהיה כפתור "מיקרופון" על מנת שהמשתמש יוכל ללחוץ עליו להתחלת הקלטה.

4.כמפתח ארצה שכאשר המשתמש יתחיל להקליט את הטקסט שלו מילות המפתח שהגדיר יהיו מותאמות בטקסט על מנת שקובץ הטקסט שלו יהיה קריא ומובן.

Epic 5 – Records & Speech Recognition:

1.כמשתמש אוכל ללחוץ על כפתור המיקרופון על מנת להתחיל לבצע הקלטות.

2.כמשתמש אוכל ללחוץ על כפתור "הרשימה" על מנת שאוכל לגשת לכל ההקלטות האחרונות שלי שנשמרו כקבצי טקסט ולערוך/ למחוק אותן.

3.כמשתמש אוכל ללחוץ על כפתור "תפריט" ומתוכו לבחור את אחת האופציות שיפתחו : הגדרות, התנתקות או מדריך למשתמש על מנת שאוכל לבצע פעולות במערכת.

4.כמשתמש אוכל ללחוץ על הכפתור "התנתקות" על מנת שאוכל להתנתק מהמערכת.

5. כמשתמש אוכל ללחוץ על הכפתור "הגדרות" על מנת שאוכל לראות את כל ההקלטות של מילות המפתח שלי ולערוך אותן במידת הצורך.

6. כמשתמש אוכל ללחוץ על הכפתור "מדריך למשתמש" על מנת שאוכל ללמוד דברים על השימוש באפליקציה.

7.כמפתח ארצה שיהיה כפתור "מיקרופון" על מנת שהמשתמש יוכל להקליט בעזרתו.

8.כמפתח ארצה שכאשר המשתמש ישתמש במיקרופון לצורך הקלטה יתבצע שימוש ב-API של גוגל על מנת שיוכל לתמלל את המילים המוקלטות.

9.כמפתח ארצה שכאשר המשתמש מסיים להקליט , התמלול שלו נשמר אוטומטית כקובץ word ונשלח אליו על מנת שיוכל לעשות בקובץ טקסט שימוש אחר כך.

10.כמפתח ארצה שכאשר המשתמש ילחץ על הכפתור "רשימה" הוא יעבור לעמוד נפרד בו תופיע רשימה עם כל קבצי הטקסט שנשמרו לו על מנת שיוכל לערוך או למחוק אותן.

11.כמפתח ארצה שכאשר המשתמש ילחץ על הכפתור של "תפריט" יפתחו בפניו 3 אפשרויות שיוכל לבחור מתוכן : הגדרות, מדריך למשתמש והתנתקות על מנת שיוכל לבצע פעולות במערכת.

12.כמפתח ארצה שכאשר המשתמש ילחץ על כפתור ההתנתקות הוא יעבור לעמוד הפתיחה של האפליקציה ויתנתק מהמערכת על מנת שיוכל להתנתק מהמשתמש שלו.

13.כמפתח ארצה שכאשר המשתמש ילחץ על כפתור "המדריך למשתמש" הוא יעבור לעמוד נפרד ובו יהיו הסברים על האפליקציה על מנת שיוכל ללמוד עליה יותר.

14.כמפתח ארצה שכאשר המשתמש ילחץ על הכפתור "הגדרות" הוא יעבור לעמוד ההגדרות האישיות שלו שם הוא יוכל לצפות בכל מילות המפתח שהגדיר על מנת שיוכל לערוך אותן.

15.כמפתח ארצה שיהיה כפתור "רשימה" על מנת שהמשתמש יוכל לצפות ברשימת קבצי הטקסט שלו.

16.כמשתמש ארצה שיהיה כפתור "תפריט" על מנת שהמשתמש יוכל לבחור אופציה אחת מתוכו ולבצע פעולות במערכת.

Epic 6 – List Of Files:

1.כמשתמש אוכל ללחוץ לחיצה ארוכה על אחד מקבצי הטקסט שיצרתי ולבחור אחת מהאופציות שיפתחו לי בתפריט : שיתוף, הורדה, מחיקה, עריכה על מנת שאוכל לבצע פעולות על הקובץ.

2.כמשתמש אוכל ללחוץ על הכפתור "שיתוף", על מנת שאוכל לשתף את קובץ הטקסט הנבחר.

3.כמשתמש אוכל ללחוץ על הכפתור "מחיקה", על מנת שאוכל למחוק את קובץ הטקסט הנבחר.

4.כמשתמש אוכל ללחוץ על הכפתור "עריכה", על מנת שאוכל לערוך את קובץ הטקסט הנבחר.

5.כמשתמש אוכל ללחוץ על הכפתור "הורדה", על מנת שאוכל להוריד את קובץ הטקסט הנבחר.

6.כמפתח ארצה שהמשתמש יוכל לצפות בכל הקבצים ששמר על מנת שהוא יוכל לערוך/למחוק/להוריד או לשתף אותם.

7.כמפתח ארצה שכאשר המשתמש ילחץ לחיצה ארוכה על אחד הקבצים יפתח בפנים תפריט עם האופציות הבאות : מחיקה, עריכה, הורדה ושיתוף על מנת שיוכל לבצע פעולות על הקבצים.

8. כמפתח ארצה שכאשר המשתמש יבחר מהתפריט "שיתוף" הקובץ ישלח אליו למייל והוא יוכל לעשות בו כרצונו על מנת שיוכל להשתמש בקובץ.

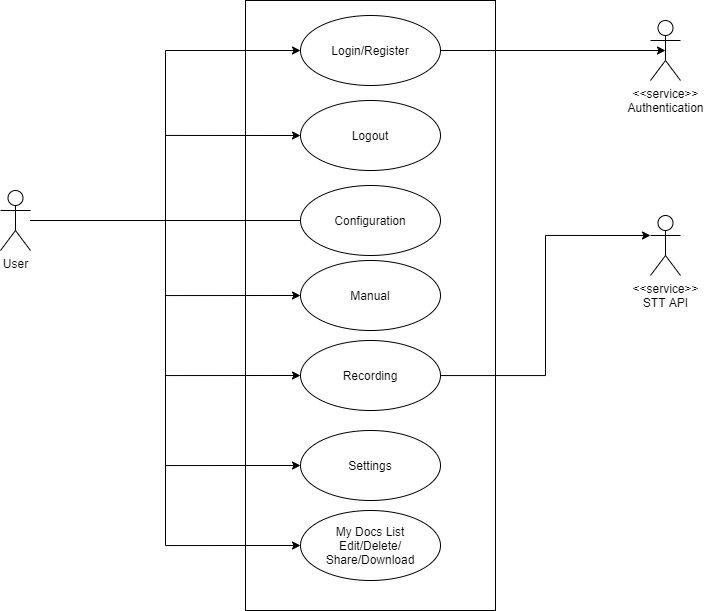
9.כמפתח ארצה שכאשר המשתמש יבחר מהתפריט "מחיקה" הקובץ ימחק מהרשימה על מנת שהמשתמש יוכל להסיר קבצים שכבר לא רלוונטיים לו.

10.כמפתח ארצה שכאשר המשתמש יבחר מהתפריט "הורדה" הקובץ יורד למכשיר שלו על מנת שיוכל להשתמש בו.

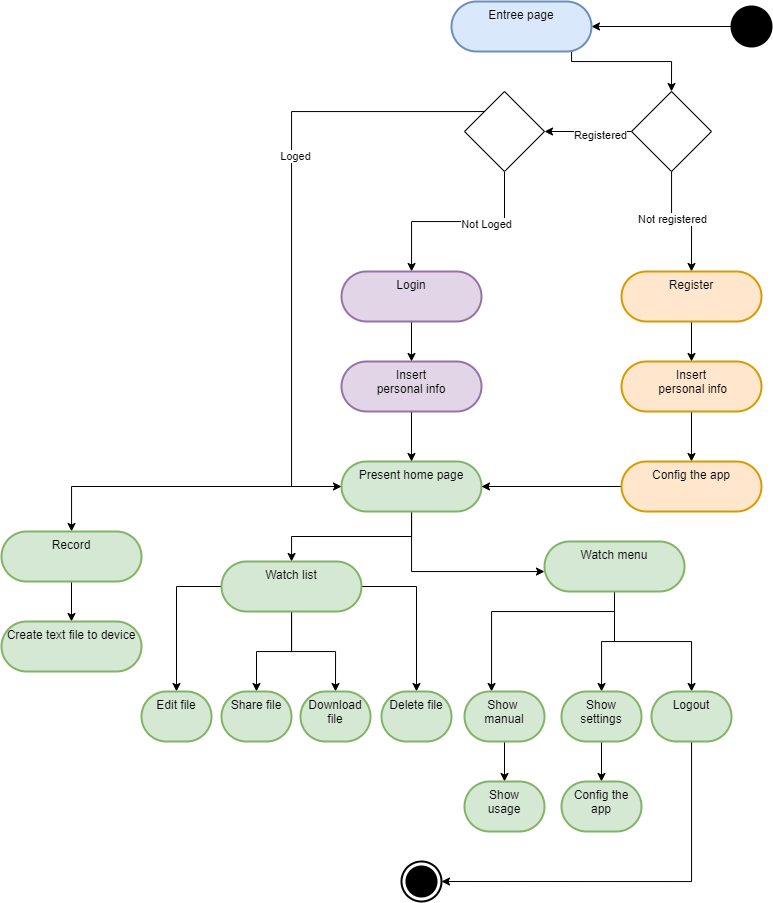
11.כמפתח ארצה שכאשר המשתמש יבחר מהתפריט "עריכה" הקובץ יפתח והוא יוכל לערוך אותו ידנית על מנת שיוכל לבצע שינויים בקובץ שלו.

**דיאגרמות**

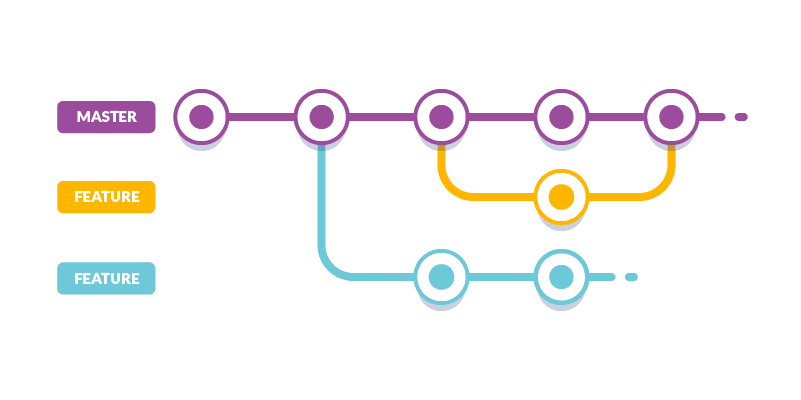
**Use Case Diagram**



**Activity Diagram**

****

**Workflow**

****

סקר ספרות

**רקע תיאורטי – מה זה Voice Speech Recognition?**

מערכת זיהוי קול/דיבור היא מערכת ממוחשבת המסוגלת לפענח [דיבור](https://he.wikipedia.org/wiki/%D7%93%D7%99%D7%91%D7%95%D7%A8) אנושי. הקלט למערכת מגיע מ[מיקרופון](https://he.wikipedia.org/wiki/%D7%9E%D7%99%D7%A7%D7%A8%D7%95%D7%A4%D7%95%D7%9F) שאליו מדבר המשתמש, משיחת [טלפון](https://he.wikipedia.org/wiki/%D7%98%D7%9C%D7%A4%D7%95%D7%9F) הנקלטת במערכת וכדומה. נהוג להבחין בין זיהוי דיבור שמטרתו לזהות את תוכן הדיבור (לצורך ביצוע פקודה או רישום טקסט) לזיהוי קול שמטרתו לזהות דובר לא ידוע, לאמת זהות דובר ידוע או לאפיין את הדיבור שלו (למשל, בעבר נעשה ניסיון להעריך האם אדם מסוים דובר שקר או אמת לפי הקול שלו). [5] טכנולוגיות זיהוי דיבור אוטומטי (ASR) כיום יכולות לזהות ולכתוב בצורה נכונה יותר מ-90% אחוז מסדרה ארוכה של מילים מדוברות עבור שפות רבות. עם זאת, אפילו אחוז גבוה זה אינו מספיק לשירותי דיבור לטקסט, שכן יש צורך ב-96+x% נכונות כדי לספק העברת הודעות מספקת. יתרה מכך, אפילו הדיוק של 90+x% בזיהוי דיבור אוטומטי אינו מתרחש מעצמו. על מנת להיות מזוהה, על הדובר לאמן את מערכת זיהוי הדיבור מראש עם מאפייני הקול והדיבור שלו. כמה מאפיינים של דיבור אזורי (דיאלקטים) מזוהים בדרך כלל רק בצורה גרועה, אפילו לאחר הכשרה מקיפה. שינויים פיזיים באיכות הקול (למשל כתוצאה משפעת) עלולים לגרום לתוצאות זיהוי גרועות יותר. הסיבה לכך היא שתהליך זיהוי הדיבור מבוסס על התאמה של פרמטרים פיזיים של אות הדיבור בפועל עם ייצוג שנוצר על בסיס מודל פונטי כללי של שפה והנתונים הפונטיים והקוליים ממפגשי האימון האישיים. אם הפרמטרים הפיזיים האישיים שונים מאלו של האימונים, ההכרה פחות מוצלחת. יתרה מכך, אם רעשי רקע מקטין את קצב האות לרעש, הדיוק עשוי לרדת מתחת ל-80 אחוז. עם זאת, מערכות זיהוי דיבור יכולות לעמוד באתגר מספר 1 (מהירות כתיבה – יוצג בהמשך) בנסיבות טובות. במקרה זה, קצב הזיהוי של ASR יהיה באופן עקרוני גבוה מספיק כדי להעביר כל מילה מדוברת לטקסט כתוב בזמן אמת. אבל יש מגבלות שצריך לקחת בחשבון. הגורם המגביל ביותר הוא שמערכות זיהוי דיבור אוטומטי אינן מסוגלות (עדיין) לזהות גבולות של ביטוי ומשפט. לכן, הפלט ממערכת זיהוי דיבור אוטומטית הוא זרם של מילים ללא כל פסיק או נקודה.[17] המטרה שלנו בפרויקט היא לייצר אפליקציה אשר יודעת להאזין לקולו של המשתמש, לזהות את המילים שיוצאות לו מהפה, להמיר אותן לטקסט עם כמה שפחות שגיאות כתיב ולשמור את זה כקובץ. זו תהיה מערכת שתתבסס על machine learning, מתחום הבינה המלאכותית שתאפשר למחשב ללמוד תוך כדי עבודה מבלי להיות מתוכנת באופן ספציפי. מטרת למידת המכונה יכולה להיות מידול, חיזוי או גילוי של עובדות לגבי העולם האמיתי. המערכת שנרצה לפתח "זיהוי דיבור קולי ותמלול " תשתמש במערכת למידת מכונה כדי ללמוד, בהינתן אותות קוליים כלשהם, מהי ההברה שיצרה אותם. [6] כיום קיימות המון מערכות שמבצעות את הפעולה ואך יודעות לתמלל בשפות שונות. למרות כל מה שקיים בשוק אנו נרצה לבנות אפליקציה משלנו המתבססת על קיימות אך נוסיף לה פיצ'רים נוספים כמו למשל נרצה לפתור את הבעיה שהצגנו קודם שמערכות זיהוי דיבור אינן מסוגלות לזהות גבולות של ביטוי ומשפט כלומר נרצה שהמערכת שלנו תדע לזהות סימני פיסוק במשפט ונשדרג אותה בעזרת פיצ'רים נוספים על מנת ליצור משהו שעוד לא נראה.

**מהו תמלול ?**

התמלול הוא כאמור העברה של הקלטה למסמך כתוב, והוא נחוץ במקרים רבים – בין אם על ידי עיתונאים אשר מקיימים ראיונות רבים ואינם יכולים לשבת מול המרואיין ולהקליד כל מילה, ולכן מקליטים את השיחה ודואגים לתמלל אותה לאחר מכן, בין אם לצרכים משפטיים על מנת לתת אינדיקציה מסוימת שבכוחה להשפיע מאוד על הסיטואציה, בין אם לצרכים ארגוניים מסוימים, למשל תמלול של ישיבות דירקטוריון אשר מהווה מעין פרוטוקול של הישיבה בצורה נוחה ומדויקת – התמלול הוא כלי עזר נהדר שבכוחו לסייע לסיטואציות רבות. עם זאת, יש לו ערך ותוקף רק כאשר הוא מעביר את הדברים שנאמרו בצורה מדויקת ומלאה, דבר שאינו פשוט לעשות. בעוד שיתרונותיו של התמלול רבים ומגוונים, לא רק שתמלול לא מוצלח לא יכול לספק אותם, הוא אף עלול להזיק בצורה ניכרת. על מנת לתמלל כראוי יש צורך במספר כישורים ומיומנויות, כגון שמיעה חדה, ידיעת השפה על בוריה (כולל אוזן טובה למבטאים וצורות דיבור שונות), בקיאות בעולם התוכן אותו מתמללים (אם מדובר, למשל, בתוכן רפואי – אי היכרות עם מונחים מהעולם הזה תקשה בצורה ניכרת על מלאכת התרגום ואף עשויה לחבל בה) וסבלנות רבה, שכן לעיתים יש צורך להקשיב שוב ושוב לאותו הקטע על מנת לפענח את הנאמר, אחרת יש תוכן חסר בתוצר הכתוב. במידה ומתמלל לא מקצועי לוקח על עצמו את אתגר התמלול, הקשיים עשויים להפוך את המשימה למפרכת במיוחד ולהימשך זמן רב, כאשר גם אז התוצאה לא בהכרח תהיה משביעת רצון. במקרה שלנו התוכנה שלנו תבצע את התמלול אוטומטית תוך כדי שהמשתמש מקליט את מבוקשו ואין צורך באדם שיקשיב להקלטה ויבצע את ההקלדה ומכאן מגיעה היעילות והחיסכון בזמן של האפליקציה שלנו.[23]

**האתגרים בהמרת דיבור לטקסט בזמן אמת**

המרת דיבור לטקסט בזמן אמת מכוונת להעביר שפה מדוברת לטקסט כתוב (כמעט) בו זמנית. זה נותן לאנשים עם לקות שמיעה, גישה לתכנים של השפה המדוברת באופן שהם למשל להיות מסוגל לקחת חלק בשיחה. תרחיש נוסף להעברת דיבור-לטקסט בזמן אמת הוא שידור חי של משחק כדורגל שבו ההערות המדוברות של הכתב מועברות כל כך מהר לכתוביות שהן עדיין מתאימות לסצנה שעליה מגיב הכתב. דוגמה מעולם השומעים תהיה דיון פרלמנטרי אשר מסתיים במסירה אלקטרונית של פרוטוקול המילה המדויקת המוצגת לעיתונאים מיד לאחר סיום הדיון. ניתן היה להמשיך בקלות את הרשימה הזו. עם זאת, רוב האנשים עם לקות שמיעה אינם מקבלים שירותי דיבור לטקסט בזמן אמת בראיונות ייעוץ, כנסים או בעת צפייה ישירה באירוע ספורט בטלוויזיה. רוב הפרוטוקולים הפרלמנטריים מוקלטים או כתובים בסטנוטייפ ומועברים לאחר מכן לטקסט קריא. מהם האתגרים של המרת דיבור לטקסט בזמן אמת שהופכים את השימוש בו לנדיר כל כך?

1. זמן - מזכירה טובה יכולה להקליד כ-300 הקשות (אותיות) בדקה. מכיוון שקצב הדיבור הממוצע הוא כ-150 מילים לדקה (עם שונות מסוימת בין הדוברים לשפות), אפילו קצב ההקלדה המקצועי בהחלט אינו גבוה מספיק כדי להעביר זרם של מילים מדוברות לצורה קריאה בזמן אמת. כתוצאה מכך, יש להגביר את מהירות ההקלדה להעברת דיבור לטקסט בזמן אמת.
2. העברת הודעות - המטרה העיקרית של העברת דיבור לטקסט היא לתת לאנשים גישה למילים מדוברות ולאירועים שמיעתיים כמעט בו-זמנית עם מימוש אירוע הצליל המקורי. עם זאת, עבור אנשים עם גישה מוגבלת לשפה המדוברת בגיל צעיר, העברת מילים מדוברות 1:1 לטקסט כתוב עשויה לפעמים לא להיות מועילה במיוחד. אם ילדים אינם נחשפים מספיק לשפה המדוברת, מערכת שפת הפה שלהם עלולה להתפתח לאט יותר ולהיות פחות יעילה בהשוואה לבני גילם. כתוצאה מכך, אנשים רבים עם לקות שמיעה מוקדמת הם פחות רגילים לכללים הדקדוקיים המיושמים בשפה בעל פה כמבוגרים ויש להם לקסיקון שכלי פחות משוכלל בהשוואה לאנשים שומעים רגילים. אם מילים אינן ידועות או אם משפטים מורכבים מדי, הצורה הכתובה אינה עוזרת להבנתן. ההשלכה של המרת דיבור לטקסט תוך לשונית היא שיש להתייחס למיומנות השפה של הקהל, כלומר יש להתאים את התמליל הכתוב ליכולות השפה של הקהל - בזמן שהדיבור נמשך. ספקי שירותי דיבור לטקסט לא רק צריכים להכיר את הקהל שלהם, הם גם צריכים לדעת אילו מילים וביטויים ניתן להחליף על ידי מקבילות שקל יותר להבין, וכיצד ניתן להפחית את המורכבות הדקדוקית.הם צריכים לדעת טכניקות כיצד להפוך את השפה כשלעצמה לנגישה יותר בזמן שהמידע המועבר נשמר. היבטים של האופן שבו ניתן להפוך את השפה לנגישה יותר יידונו בחלק הבא "עיבוד טקסט".
3. הצגה בזמן אמת של הטקסט הכתוב - קריאה פירושה בדרך כלל שמילים כבר רשומות. כשהם מוצגים עם טקסט כתוב, אנשים יקראו במהירות הקריאה האישית שלהם. עם זאת, הדבר אינו אפשרי בהמרת דיבור לטקסט בזמן אמת. כאן, הטקסט נכתב ונקרא כמעט בו זמנית, והשליטה על מהירות הקריאה עוברת לפחות חלקית אל הדובר וספק הדיבור לטקסט. הטקסט אינו קבוע מראש, במקום זאת מייצרים מילים חדשות באופן רציף והקוראים חייבים לעקוב מקרוב אחר תהליך ייצור המילים הזה אם הם רוצים להשתמש ביכולות בזמן אמת של העברת דיבור לטקסט. בגלל אינטראקציה זו של כתיבה וקריאה, יש להתאים בצורה מיטבית את הצגת הטקסט הכתוב לצרכי הקריאה של הקהל.

לסיכום האתגרים של המרת דיבור לטקסט בזמן אמת הם : להיות מהיר מספיק בהפקת שפה כתובה, לעמוד בציפיות של הקהל ביחס למאפיינים של טקסט כתוב, הצגת המילים הכתובות בצורה אופטימלית לזיהוי ומובן עבור הקוראים.[17]

**שיטות להמרת דיבור לטקסט בזמן אמת**

ישנן שלוש שיטות שניתן לבצע בעת מימוש (כמעט) העברת דיבור לטקסט בזמן אמת: זיהוי דיבור, רישום הערות בעזרת מחשב (CAN) וגישה לתקשורת (או בעזרת מחשב) תרגום בזמן אמת (CART). השיטות שונות: 1. ביכולתם להפיק תמלילים מדויקים בזמן אמת. 2. מתייחסות לתנאים שבהם ניתן ליישם כהלכה שיטות אלו. 3. מתייחסות לכמות ההכשרה הדרושה כדי להפוך לספק שירותי דיבור לטקסט טוב.

**במה המערכת שלנו יכולה להועיל למשתמש?**

המשתמש יוכל לחסוך לעצמו זמן, כאשר בימינו זהו מצרך הכרחי .זמן ההקלדה של המילים יחסך לו ומה שיצטרך לעשות זה רק ללחוץ על כפתור ולדבר, כמובן לא במהירות גבוהה מדי שהמערכת תוכל להבין את המילים היוצאות מפיו. מעבר לכך המשתמש יוכל להשתמש במערכת לאו דווקא במחשב אלא גם smartphone, שעון חכם, iPad וכדומה, מה שהופך את השימוש במערכת לנגיש יותר גם מחוץ לבית.

**מטרת הפרויקט**

מטרתנו העיקרית היא בעצם ליצור מערכת זיהוי דיבור קולי ותמלול ולשדרג אותה בפיצ'רים נוספים שלא קיימים היום בשאר המערכות הדומות שקיימות, בנוסף נרצה שהתמלול ישמר למשתמש כקובץ כאשר הוא יסיים. נרצה שהמערכת תהיה נגישה לכלל האוכלוסייה ותשפר את איכות חייהם בחסכון בזמן בעבודה במחשבים ובמכשירים חכמים כמו למשל הקלדה במנועי חיפוש, כתיבת מסמכים ארוכים, כתיבת הודעה ועוד. בנוסף כל משתמש יתבקש לבצע לפני השימוש הגדרה של מילות מפתח המתאימות ליצירת פורמט אישי על ידי זיהוי המילים המוגדרות על ידיו למשל המילה end תומר על ידי המערכת לסימן הפיסוק "נקודה" (.).

**Machine Learning**

למידת מכונה (Machine Learning או בקיצור ML) היא תת-תחום במדעי המחשב והבינה המלאכותית (AI) העוסק במגוון משימות חישוביות בהן התכנות הקלאסי לא ישים. המאפיין המרכזי של למידת מכונה הוא בפיתוח אלגוריתמים שאינם מבוססים על סט חוקים מוגדר מראש, אלא לומדים מתוך מצבור דוגמאות. המטרה המרכזית של למידת מכונה היא טיפול ממוחשב בנתונים, על ידי מידול, חיזוי או גילוי עובדות מן העולם האמיתי, עבור בעיות שלא ניתן לכתוב להן תכנת מחשב "קלאסית". כלומר, אם בתכנות הקלאסי כתיבת התוכנה מתבססת על לוגיקה ברורה שמורכבת מהגדרת תנאים מסוימים (בהם לולאות, דרכי פעולה במצבים שונים ועוד), אזי למידה מכונה (Machine Learning) מאופיינת [באלגוריתמים](https://www.bdo.co.il/he-il/%D7%91%D7%9C%D7%95%D7%92%D7%99%D7%9D/bdo-tech-hub/%D7%9E%D7%A8%D7%A5-2019/%D7%90%D7%9C%D7%92%D7%95%D7%A8%D7%99%D7%AA%D7%9D-(algorithm))שמטרתם לאפשר למחשב ללמוד את התנאים ולהסיק מידע באופן עצמאי מתוך כמות גדולה של דוגמאות. יכולת זו יעילה מאוד בסיטואציות מורכבות בהן קשה להגדיר מראש סט חוקים או תנאים- מנהיגה אוטונומית, דרך אבחון גידולים סרטניים, ועד לחיזוי העדפות משתמש. [15] למידת מכונה כוללת אוסף ידוע של כמה עשרות אלגוריתמים ניתן לסווג אלגוריתמים אלו על פי מספר תבחינים :**1. עצלנים (lazy ) מול חרוצים (eager )–** אלגוריתם עצלן לא מבצע חישוב עד שלא מתבקש לענות על שאלה, כאשר נשאלת שאלה האלגוריתם אוסף נתונים רלוונטיים, מבצע חישוב ונותן תשובה. לעומת זאת האלגוריתמים החרוצים משתמשים בנתוני הלימוד על מנת לבנות מודל מתמטי אשר מהווה בסיס לפתרון הבעיה. כאשר האלגוריתם מתבקש לענות על שאלה הוא משתמש בנתוני השאלה על מנת לפתור את המודל. [6] **2. מקומי (local) מול גלובלי (global)-**אלגוריתם מקומי הוא אלגוריתם שמחשב פתרון שתקף רק לסביבה שבה נשאלה השאלה. אלגוריתם גלובלי לעומתו מחשב מודל שאמור להיות תקף בכל מרחב הנתונים הרלוונטי.[6]

**להלן מספר אלגוריתמים נפוצים:**

* אלגוריתם שכן קרוב
* עץ החלטה לומד
* רגרסיה מקומית
* רשת עצבית מלאכותית
* רשתות ביסיאניות
* אשכולות מבוססי מרחק
* אלגוריתם ציפייה – מקסום
* אלגוריתם גנטי
* אלגוריתם של לימוד מותקן (boosting ,bagging)
* חוקים אסוציאטיביים
* PSO- אופטימיזציה מבוססת תבונת נחיל
* ACO – אופטימיזציה מבוססת תבונת נמלים
* מכונת וקטורים תומכים

**האלגוריתמים בהם נשתמש**

כמו שכבר הזכרנו קודם לתחום של למידת המכונה יש אלגוריתמים רבים שנעשה בהם שימוש על מנת להגיע ללימוד הטוב ביותר למכונה ושתעבוד בצורה המדויקת ביותר. בפרויקט שלנו אנחנו עובדים עם משהו קיים ומשפרים אותו ולכן נשתמש ב-API של גוגל, בשל כך אנחנו לא יכולים לדעת בדיוק באיזה אלגוריתמים נעשה שימוש . באופן כללי אנו יודעים שיש שימוש ב-Machine Learning (ML) וב- Artificial intelligence (AI). מכאן אנו סבורים שהם השתמשו באלגוריתמים כמו: Natural Language Processing – NLP שתפקידו לעסוק בבעיות הקשורות לעיבוד ומניפולציה על שפה טבעית והבנה של שפה טבעית על מנת לגרום למחשבים "להבין" דברים שנאמרים נכתבים בשפות אנושיות. [14] ואלגוריתמים דומים נוספים..

**בינה מלאכותית AI (Artificial Intelligence)**

בינה מלאכותית היא שם מטאפורי למצב שבו מנסים לדמות את יכולות החשיבה האנושית באמצעים טכנולוגים, למשל, באמצעות [למידת מכונה](https://he.wikipedia.org/wiki/%D7%9C%D7%9E%D7%99%D7%93%D7%AA_%D7%9E%D7%9B%D7%95%D7%A0%D7%94) .התחום הוא ענף של [מדעי המחשב](https://he.wikipedia.org/wiki/%D7%9E%D7%93%D7%A2%D7%99_%D7%94%D7%9E%D7%97%D7%A9%D7%91) ,העוסק ביכולת לתכנן חומרה או תוכנה ספרתית קרי של [מחשבים](https://he.wikipedia.org/wiki/%D7%9E%D7%97%D7%A9%D7%91) לפעול באופן המציג יכולות המתחזות חלקית להתנהגותה של [הבינה האנושית](https://he.wikipedia.org/wiki/%D7%97%D7%95%D7%9B%D7%9E%D7%94). המבחן המקובל ביותר לבינה מלאכותית הוטבע בשנת [1950](https://he.wikipedia.org/wiki/1950)  על ידי [אלן טיורינג](https://he.wikipedia.org/wiki/%D7%90%D7%9C%D7%9F_%D7%98%D7%99%D7%95%D7%A8%D7%99%D7%A0%D7%92), וידוע בשם" [מבחן טיורינג](https://he.wikipedia.org/wiki/%D7%9E%D7%91%D7%97%D7%9F_%D7%98%D7%99%D7%95%D7%A8%D7%99%D7%A0%D7%92) :"מכונה תחשב לתבונית, אם יינתן לאדם, "הבוחן" היושב בחדר סגור, לנהל שיחה באמצעות ממשק מחשב (Console) עם שתי ישויות שנמצאות בחדר השני, כאשר אחת מהן תהיה מכונה והשנייה אנושית, והמשוחח לא יוכל לזהות מי משתי הישויות היא מכונה או אדם מכונה.[16]

**Google API**

API = Application Programming Interfaces . ממשקי Google API הם ממשקי תכנות יישומים שפותחו על ידי גוגל המאפשרים תקשורת עם שירותי גוגל ושילובם עם שירותים אחרים. דוגמאות לכך כוללות חיפוש, gmail , גוגל תרגום או מפות גוגל. יישומי צד שלישי יכולים יכולים להשתמש בממשקי API אלה כדי לנצל או להרחיב את הפונקציונליות של השירותים הקיימים. ממשקי ה-API מספקים פונקציונליות כמו ניתוח, למידת מכונה כשירות או גישה לנתוני משתמש. שימוש בכל ממשקי ה-API דורש אימות והרשאה באמצעות פרוטוקול Qauth 2.0 . זהו פרוטוקול פשוט. כדי להתחיל, יש צורך להשיג אישורים ממסוף המפתחים. לאחר מכן אפליקציית הלקוח יכולה לבקש גישה משרת ההרשאות של גוגל ואז יכולה לגשת לשירותי Google API. גוגל שיפרה את זיהוי הדיבור שלה על ידי שימוש בטכנולוגיה חדשה ביישומים רבים עם Google App כגון Google411, חיפוש קולי בנייד, פעולות קוליות, קלט קולי למקלדת, ממשקי API למפתחי Android, חיפוש קולי במחשב שולחני, תמלול YouTube ותרגום, ניווט,TTS. שימוש ב-Google API יביא לנו את התוצאות המדויקות ביותר .

**המודלים במערכת**

**מודל משתמש** – מודל המשתמש יכיל בתוכו את המילים השמורות שיתבקש להגדיר מראש לפני תחילת ההקלטה, מילים אלו יהוו עבורו מעין מילון כאשר ירצה להשתמש בסימני הפיסוק השונים בקובץ שלו. המשתמש לא יוכל להתחיל את השימוש באפליקציה ללא הגדרת מילים אלו. למשל: המשתמש יגדיר שכאשר המערכת תשמע את המילה "new line " באנגלית היא תשמור אותה במילון של אותו משתמש וכאשר היא תשמע את המילה "new line " תתבצע ירידת שורה בקובץ שיתקבל בסיום ההקלטה וכך הלאה .

**מודל מכונה** – במודל המכונה לאחר שהמשתמש יקליט את הקטע אותו ירצה להפוך לטקסט ההקלטה שנקלטה במערכת תיכנס אל תוך המכונה ושם תעבור תהליך עיבוד והמכונה תמיר אותה לקובץ בפורמט שיבחר על ידי המשתמש (pdf ,word).

**המצב הקיים בשוק**

כיום קיימות המון אפליקציות עם תפקיד זהה לזו שאנחנו רוצים לפתח . כולן מבצעות את אותה פעולה, זיהוי קולי ותמלול אך לכל אחת יתרונות וחסרונות משלה נרחיב על כמה מהן:

1**. Dragon Professional**– נקראת גם dragon for PC, אפליקציה שיכולה לשמש כל אחד מתלמידים/סטודנטים ועד לאנשים מקצועיים ולעסקים. האפליקציה יכולה לעזור במשימות יום יומיות כמו: כתיבת שיעורי בית, שליחת מיילים ואפילו גלישה באינטרנט בעזרת כך שהמשתמש מדבר והיא כותבת. ידועה בשל יכולות ההכתבה והתמלול המהירות שלה ומסייעת לאנשים עובדים ביצירת מסמכים.[8]  
  
**2.Google Now –** תוכנה של מנוע החיפוש של גוגל (google search), זמינה למכשירי android ו-IOS . התוכנה משולבת היטב עם מערכת ההפעלה של android כך שניתן להשתמש בה לביצוע של כל פונקציונליות הקיימת בה, זמינה גם במחשבים אישיים כחלק מדפדפן האינטרנט google chrome.   
במכשירי אנדרואיד ניתן להשתמש ב-google now לקבלת שיחות, שליחת הודעות טקסט ופתיחה וסגירה של האפליקציה. עבור מכשירי IOS ניתן להשתמש בה לצורך חיפוש פונקציונלי. הפעלת התכונה מתבצעת על ידי הקלקה או נגיעה בסמל הקטן של המיקרופון, בצד חלון החיפוש או, בטלפונים ניידים עם גרסת "[אנדרואיד](https://he.wikipedia.org/wiki/%D7%90%D7%A0%D7%93%D7%A8%D7%95%D7%90%D7%99%D7%93_(%D7%9E%D7%A2%D7%A8%D7%9B%D7%AA_%D7%94%D7%A4%D7%A2%D7%9C%D7%94)) " מתאימה, ישירות על ידי הפקודה הקולית: OK Google כאשר התכונה מופעלת יופיע חיווי מתאים ואז היישומון מאזין לשאלות של המשתמש. תכונה זו אמורה להיות שימושית במיוחד במצבים בהם הידיים תפוסות, כמו למשל בעת בישול במטבח, עבודה עם כלים מכניים או אפילו, בזמן נהיגה.[9]

**3.Siri –** סירי היא מערכת בינה מלאכותית אוניברסלית מסוג עוזרת אישית ווירטואלית למכשירי אפל הנתמכת על ידי 21 שפות שונות. התוכנה מותקנת מראש במכשירי אפל ומגיבה בקולה של סירי. המערכת מעבדת [שפה טבעית](https://he.wikipedia.org/wiki/%D7%A9%D7%A4%D7%94_%D7%98%D7%91%D7%A2%D7%99%D7%AA) [מדוברת](https://he.wikipedia.org/wiki/%D7%93%D7%99%D7%91%D7%95%D7%A8), כדי לענות על שאלות, לתת המלצות, לבצע פעולות בהתאם לבקשות המשתמש ולענות על שאלות באמצעות משאבי המידע [וולפרם אלפא](https://he.wikipedia.org/wiki/%D7%95%D7%95%D7%9C%D7%A4%D7%A8%D7%9D_%D7%90%D7%9C%D7%A4%D7%90) ו-[וויקיפדיה](https://he.wikipedia.org/wiki/%D7%95%D7%99%D7%A7%D7%99%D7%A4%D7%93%D7%99%D7%94). מאפייני המערכת כוללים הסתגלות להעדפותיו של המשתמש לאורך זמן והצגת תוצאות אישיות, בנוסף ליכולת להזמין מונית או אפילו להזמין מקום במסעדה לארוחת ערב . סירי יכולה לבצע שיחות, לשלוח הודעה, לבצע חיפוש באינטרנט, להגדיר אזעקות, טיימרים ותזכורות. [10]

**4. Amazon Lex –** שירות לבניית ממשקי שיחה לכל יישום באמצעות קול וטקסט. התוכנה מספקת את הפונקציונליות המתקדמת של למידה עמוקה של זיהוי דיבור אוטומטי להמרת דיבור לטקסט והבנת שפה טבעית כדי לזהות את כוונת הטקסט, לאפשר למשתמש לבנות אפליקציות עם חוויות משתמש טובה ושיחה אמיתית. [11] האפליקציה ידועה בכינוייה אלכסה, אלכסה היא סייעת וירטואלית שמפותחת על ידי אמזון. אלכסה מסוגלת לנהל שיחות, להשמיע מוזיקה, להכין רשימות, [להגדיר התראות](https://he.wikipedia.org/wiki/%D7%A9%D7%A2%D7%95%D7%9F_%D7%9E%D7%A2%D7%95%D7%A8%D7%A8), להזרים פודקאסטים, להשמיע ספרים קוליים, להציג את תחזית מזג האוויר, מצב התנועה, ספורט, ועוד מידע בזמן אמת, כגון [חדשות](https://he.wikipedia.org/wiki/%D7%97%D7%93%D7%A9%D7%95%D7%AA_(%D7%AA%D7%A7%D7%A9%D7%95%D7%A8%D7%AA)). אלכסה יכולה גם לשלוט במספר [מכשירים חכמים](https://he.wikipedia.org/w/index.php?title=%D7%9E%D7%9B%D7%A9%D7%99%D7%A8%D7%99%D7%9D_%D7%97%D7%9B%D7%9E%D7%99%D7%9D&action=edit&redlink=1) באמצעות מערכת [בית חכם](https://he.wikipedia.org/wiki/%D7%91%D7%99%D7%AA_%D7%97%D7%9B%D7%9D). רוב המכשירים שעליהם פועלת אלכסה מאפשרים למשתמש להפעיל את הסייעת הקולית על ידי אמירת מילה (למשל :"אלכסה").[12]

**סוגי אפליקציות**

**Mobile Applications**

אפליקציית מובייל או יישום מובייל היא יישום מחשב המיועד לשימוש בטלפונים חכמים (סמארטפונים), מחשבי לוח (טאבלטים) ומכשירים ניידים מסוגים אחרים. היישומונים זמינים בדרך כלל דרך פלטפורמות להפצת יישומים, אשר לרוב מתופעלות על ידי ה[חברות](https://he.wikipedia.org/wiki/%D7%97%D7%91%D7%A8%D7%94_(%D7%AA%D7%90%D7%92%D7%99%D7%93)) שבבעלותן [מערכות הפעלה](https://he.wikipedia.org/wiki/%D7%9E%D7%A2%D7%A8%D7%9B%D7%AA_%D7%94%D7%A4%D7%A2%D7%9C%D7%94) למכשירים ניידים, לדוגמא: App Store של חברת אפל, Google Play של גוגל, Windows Phone Store של מייקרוסופט ו-BlackBerry App World של בלקברי. בדרך כלל מורידים את האפליקציות ישירות למכשירי היעד, כדוגמת מכשירי אייפון, אנדרואיד או windows phone, אבל לפעמים ניתן להוריד את האפליקציות גם למחשבים אישיים ולפטופים. במקור, אפליקציות מובייל פותחו למטרות עבודה וקבלת מידע, כדוגמת אפליקציות דואר אלקטרוני, יומן אלקטרוני, אנשי קשר, מידע על מניות ומזג אוויר. עם זאת, הביקוש מצד המשתמשים והזמינות של כלי פיתוח הובילו להתרחבות מהירה גם לתחומים אחרים כגון משחקי מחשב, שירותים מבוססי מיקום ו-GPS, שירותי בנקאות, מעקב אחר הזמנות ורכישת כרטיסים. הגידול המהיר במספר ובמגוון של האפליקציות הוביל ליצירה של מגוון רחב של מקורות עם סקירות והמלצות של אפליקציות כגון בלוגים, כתבי עת ושירותים מקוונים ייעודיים לגילוי אפליקציות.[18] בעידן המודרני של מערכות מידע ותקשורת, אנשים רגילים להשתמש במחשב ובאפליקציות מחשב. אבל השימוש והפיתוח של יישומים ניידים הם מגזר חדש וצומח במהירות. יש השפעה חיובית גלובלית של אפליקציה לנייד. השימוש באפליקציות סלולריות במדינה מפותחת הופכים לקל. כלומר, קל לשימוש ונגיש מכל מקום. כיום, כל כך הרבה אנשים משתמשים באפליקציה סלולרית כדי ליצור קשר עם חברים, לגלוש באינטרנט, ניהול תוכן קבצים, יצירה וטיפול במסמכים, בידור ועוד. מכל מקום המשתמש יכול לקבל התקנה של אפליקציה לנייד. לא רק לאפליקציית הנייד יש השפעה על המשתמש אלא גם היא ממלאת תפקיד חשוב בעסקים. חברות עסקיות רבות מרוויחות הכנסות באמצעות אפליקציה לנייד. לאפליקציה הסלולרית יש השפעה על החברה. האפליקציה לנייד פועלת בסביבה ניידת שהשימושיות שלה תלויה במספר גורמים כגון: רזולוציית מסך, מגבלות חומרה, שימוש יקר בנתונים, בעיות קישוריות, אפשרויות אינטראקציה מוגבלות. בשנים האחרונות חברות הסלולר מנסות לפתח מכשירים ניידים עם יותר רזולוציית מסך, יותר אחסון, קישוריות טובה יותר המספקים סביבה טובה יותר עבור אפליקציות ניידות מודרניות.[19]

**native applications**

אפליקציות native מותקנות על המכשיר עצמו ומפותחות במיוחד עבור מערכת הפעלה סלולרית מסוימת. אפליקציות אלו זמינות בחנויות אפליקציות כגון Apple App Store, Google Play Store וכו'. אפליקציות native פועלות במערכות הפעלה ספציפיות לנייד כגון Apple iOS או Android OS. אם האפליקציה מיועדת למערכת ההפעלה אנדרואיד היא לא תעבוד ב-Apple iOS או Windows OS. עלינו לבנות אפליקציות נפרדות לכל מערכת הפעלה אם ברצוננו להפעיל את האפליקציה שלנו בכל מערכות ההפעלה העיקריות. המשמעות היא שעלינו להוציא יותר כסף ויותר מאמץ (זמן, משאבים).[20]

**יתרונות:**

* אפליקציות native מהירות יותר מאפליקציות web.
* אפליקציות native יכולות לגשת למשאבי מערכת/מכשיר כגון GPS או מצלמה.
* אפליקציות אלו יכולות לעבוד ללא חיבור לאינטרנט.
* לאפליקציות אלה יש יותר בטיחות ואבטחה מאשר אפליקציות אינטרנט (web) שכן אפליקציות native חייבות להיות מאושרות על ידי ה-App Store .
* קל יותר לפתח יישומים אלה בשל הזמינות של כלי מפתחים, רכיבי ממשק ו-SDKs.

**חסרונות:**

* אפליקציות אלו יקרות יותר לפיתוח בהשוואה לאפליקציות אינטרנט.
* עיצוב ובניית האפליקציה עבור פלטפורמות שונות כמו iOS ואנדרואיד.
* יקרים ולוקחים זמן.
* שמירה ועדכון עקבי של אפליקציות native גורמות ליותר עלויות.
* קשה לקבל אפליקציה native לנייד שאושרה על ידי ה-App Store.

**Web Applications**

ניתן לגשת לאפליקציות web דרך דפדפן האינטרנט של המכשיר הנייד, אפליקציות web מבוססות על יישומים התומכים באינטרנט. על מנת לגשת איננו צריכים להוריד ולהתקין את האפליקציה במכשיר נייד. האפליקציה מפותחת כדפי אינטרנט ב-HTML ו-CSS, חלקים אינטראקטיביים מפותחים ב-JQuery, JavaScript או שפה דומה. ניתן להשתמש באפליקציית web הבודדת ברוב המכשירים המסוגלים לגלוש באינטרנט, זה לא תלוי במערכת ההפעלה שבה הם משתמשים. [20]

**יתרונות:**

* אפליקציות web פועלות בדפדפן כך שאין צורך להתקין או להוריד אותן.
* קלות לתחזוקה מכיוון שיש להן בסיס קוד משותף ללא קשר למערכת ההפעלה.
* ניתן להגדיר אפליקציות אלו כך שיעדכנו את עצמן או באופן אוטומטי.
* אפליקציות web קלות ומהירות יותר לבנייה מאשר אפליקציות native לנייד.
* ב-Web Apps לא נדרש אישור בחנות האפליקציות כך שניתן להפעיל אותן בקלות.

**חסרונות:**

* אינן פועלות ללא חיבור לאינטרנט.
* פועלות לאט יותר מאפליקציות לנייד.
* קשה לגלות אפליקציות web מכיוון שהן אינן מתארחות במסד נתונים ספציפי כמו חנות אפליקציות.
* לאפליקציות web יש סיכונים גבוהים יותר ואיכות ירודה.

**Hybrid Applications**

אפליקציה היברידית הינה אפליקציה אשר פותחה באמצעות קוד יחיד, אשר מסוגלת לרוץ במספר מערכות הפעלה (אנדרואיד, iOS, windows ועוד). בשונה מאפליקציות native בהן שפת הפיתוח תלויה במערכת ההפעלה (למשל: פיתוח אפליקציה לאנדרואיד יתבצע ב-java ואילו פיתוח אפליקציה ל-iOS יתבצע ב-Swift/Objective C) פיתוח אפליקציה היברידית מתבצע על ידי שפות web, למשל: HTML5,CSS3,JAVASCRIPT אשר עוברות בסוף תהליך הפיתוח התאמה למספר מערכות הפעלה באמצעות Frame Works. פיתוח אפליקציה בשיטה היברידית מאפשר לפתח אפליקציה פעם אחת למספר מערכות הפעלה.[21]

**יתרונות:**

* פיתוח מהיר יותר – במקום לפתח את אותה האפליקציה בשפות שונות בפיתוח native, בפיתוח היברידי ניתן לפתח את האפליקציה פעם אחת בלבד. מעבר לכך, פיתוח אפליקציה היברידית יהיה מהיר יותר היות ומדובר בפיתוח פחות מורכב ובשפות web הפשוטות בהרבה ביחס לשפות כמו: Java, Objective C, Swift.
* פיתוח זול יותר – במקום לפתח את אותה האפליקציה פעמיים, אנו נדרשים לפתחה פעם אחת בלבד ולכן העלויות נמוכות יותר. נוסף על כן, עלות העסקת מפתחי אפליקציות ל-IOS או Android גבוהה בהרבה ביחד למתכנתי web וזה בא לידי ביטוי במחיר הפיתוח.
* תחזוקה קלה ומהירה – במקום לבצע תיקונים ו/או הוספת פיצ'רים חדשים ב-2 סביבות שונות, בפיתוח היברידי השינויים מתבצעים בסביבה אחת בלבד עבור כל הפלטפורמות.

**חסרונות:**

* שימוש בכל תכונות המכשיר – כשאנו מפתחים אפליקציה היברידית, הפיתוח מוגבל ואין אנו יכולים להשתמש בפונקציונליות המלאה של המכשיר. זה לא אומר שאי אפשר להשתמש הרכיבים כמו GPS , מצלמה וכו'. אפשר לעשות הרבה דברים ולהשתמש במרבית הרכיבים והתכונות, השאלה היא מה הדרך הנכונה ביותר.
* ביצועים פחות טובים ביחס לפיתוח native – כאשר אנו מפתחים בשפה המקורית של המכשיר, הביצועים יהיו טובים יותר.
* חנויות – תהליך העלאת אפליקציות לחנויות הינו תהליך קריטי והחשוב ביותר בהשקת אפליקציה. מדובר בתהליך לא פשוט בכלל, המחייב עמידה בתנאים ובמדיניות של חנויות הסטור. כשמדובר באפל קשה מאוד להעלות ל-App Store אפליקציות היברידיות אך זהו תהליך בר ביצוע כאשר מפתחים את האפליקציה בצורה נכונה.

**מסד נתונים – firebase**

Firebase היא פלטפורמה ליצירת אפליקציות עם שירותי ענן שפותחה על ידי גוגל ומשמשת לפיתוח יישומי רשת ואפליקציות מובייל. היא מספקת כלים למעקב אחר ניתוח נתונים, דיווח ותיקון קריסות אפליקציה ויצירת שיווק. [22]  
Firebase מציע מספר שירותים:  
1.Analytics – מציע דיווח חינם וללא הגבלה על עד 500 אירועים נפרדים. מציג נתונים על התנהגות משתמשים באפליקציות ios ו-Android ומאפשר קבלת החלטות טובה יותר לגבי שיפור ביצועים ושיווק אפליקציות.   
2.Authentication – אימות Firebase מקל על מפתחים לבנות מערכות אימות מאובטחות ומשפר את חווית ההתחברות והשימוש למשתמשים. תוכנה זו מציעה פתרון זהות מלא, תומך בחשבונות דוא"ל וסיסמא.  
3.Cloud Messaging – העברת הודעות בענן (FCB- Firebase Cloud Messaging) הוא כלי העברת הודעות חוצה פלטפורמות המאפשר לחברות לקבל ולמסור באופן אמין הודעות ב-ios, אנדרואיד ובאינטרנט ללא עלות.   
4.Realtime Database – מסד נתונים מסוג NoSQL המאפשר אחסון וסנכרון נתונים בין משתמשים בזמן אמת. הנתונים מסונכרנים בין כל הלקוחות בזמן אמת ועדיין זמינים כאשר האפליקציה לא פעילה.  
5.Crashlytics – כתב קריסות בזמן אמת שעוזר למפתחים לעקוב, לתעדף ולתקן בעיות יציבות שמפחיתות את איכות האפליקציות שלהם. עם Crashlytics מפתחים מבלים פחות זמן בארגון ופתרון תקלות ויותר זמן בבניית תכונות עבור האפליקציות שלהם.  
6.Performance – שירות ניטור הביצועים של Firebase נותן למפתחים תובנות לגבי מאפייני הביצועים של אפליקציות ios ואנדרואיד שלהם כדי לעזור להם לקבוע היכן ומתי ניתן לשפר את הביצועים של האפליקציות שלהם.   
7.Test lab – תשתית לבדיקת אפליקציות מבוססת ענן. בפעולה אחת מפתחים יכולים לבדוק את יישומי ios או אנדרואיד שלהם במגוון מכשירים. הם יכולים לראות את התוצאות, כולל סרטונים, צילומי מסך ויומנים במסוף Firebase.

בחרנו במסד נתונים זה מכיוון שהוא יכול לטפל בכמות רבה של נתונים, מתממשק בצורה טובה עם Flutter והוא חינמי.

**סוגי בדיקות**

כיום, אין מוצר, אפליקציה, אתר אינטרנט או כל אפיק דיגיטלי אחר שלא עובר בדיקה על ידי צוות QA מוסמך ואיכותי. אלו הן 8 סוגי הבדיקות שקיימות בעולם בדיקות התוכנה:

1. Usability/ בדיקות שימושיות – בבדיקה זו מוודאים עד כמה נוחות השימוש איכותית, האם העיצוב יעיל ונוח, האם קל להשתמש בו והאם רמת חווית המשתמש גבוהה.
2. Accessibility/ בדיקות נגישות – כמה המוצר נגיש עבור בעלי מוגבלויות, כידוע, יצא חוק שמחייב כל בעל עסק או ארגון שהמוצרים שלו יהיו נגישים עבור עיוורים, כבדי שמיעה ושאר מוגבלויות אחרות.
3. Functional/ בדיקות פונקציונליות – בדיקות אלו נוצרו לצורך אימות פעילות המערכת. בכל פעם שמייצרים או מפתחים מוצר, ישנו מסמך דרישות המצורף אליו. בגדול המטרה של בדיקה זו היא לוודא שהמוצר אכן נענה לכל הדרישות שהתבקשו ושהמוצר עושה את מה שהוא צריך לעשות. זוהי למעשה אחת הבדיקות החשובות ביותר שמלמדים בקורס QA.
4. Non Functional/ בדיקות לא פונקציונליות – בדיקה כיצד המערכת מתפקדת תחת עומסים שונים ומה רמת הביצועים שלה. מה קורה כשיש המון משתמשים על אותה פלטפורמה? קיימים מספר כלים שונים כמו: WebLoad, teamSystem, LoadRunner שיכולים לדמות מספר רב של משתמשים על אותו מוצר, אפליקציה או אתר ובכך לבחון את התפקוד שלו.
5. Internationalization and Localization/ בדיקות בינלאומיות – קיימות המון תוכנות ואתרים גלובליים הפועלים בכמה שפות שונות. נניח כמו ויקיפדיה או waze . בבדיקה זו צריך לוודא שכל ההליך של התרגומים או המעבר בין השפות בהחלט תקין, ושאין שגיאות בין עמודים שונים בשפות שונות.
6. GUI/ בדיקות ממשק לקוח – בדיקה מקיפה שבודקת את הפקדים והשדות במסך. האם הפורמט של השדות תקינים ופעילים כמו שצריך. כל זאת כמובן בהתאם לחוקיות המוגדרת ברמת המסך. בגדול מדובר על שדות חובה שיש למלא, מה המידע שניתן למלא שם ולכמה תווים ניתן להגיע (למשל סיסמא שאמורה להיות עד 6 תווים הכוללים ספרות ואותיות).
7. Sanity & Smoke/ בדיקות שפיות ועשן – מדובר על בדיקות מאוד בסיסיות, שיכולות להעניק את המידע האם ניתן להיכנס לבדיקות מעמיקות יותר או לא. במידה והן אכן עוברות בהצלחה, זה לא מעיד על איכות או על כך שהמוצר תקין, כיוון שנותר עוד מספר בדיקות מעמיקות שיש להיערך אליהן.
8. Compatibility/ בדיקות תאימות – כמעט וכל מוצר שמפותח היום הוא web, לכן חובה שתהיה לו תאימות להמון דפדפנים שונים, הן על הdesktop והן על המובייל. בעזרת בדיקות אלו מוודאים שהליך זה באמת מתרחש. כך שלא יקרה תסריט שבו לא תהיה תאימות לדפדפן, מערכת ההפעלה או תוכנות אחרות שאמורות להתממשק איתן.

# טבלת עקיבות

|  |  |  |
| --- | --- | --- |
| Requirement # | Requirement Text | Test Case ID |
| Epic 1- #1 | כמשתמש אוכל ללחוץ על כפתור "התחברות" או "הרשמה" על מנת שאוכל להתחיל לבצע שימוש במערכת. | Entry\_page\_1 |
| Epic 1- #2 | כמפתח ארצה שעמוד הפתיחה של האפליקציה יכיל 2 כפתורים "התחברות" ו-"הרשמה" על מנת שהמשתמש יוכל לבחור מבניהם ולהתחיל להשתמש במערכת. | Entry\_page\_2 |
| Epic 1- #3 | כמפתח ארצה שכאשר המשתמש ילחץ על אחד מן הכפתורים "התחברות" או "הרשמה" הוא יעבור מיד לעמוד המתאים על מנת שיוכל להמשיך את השימוש באפליקציה. | Entry\_page\_3 |
| Epic 2- #1 | כמשתמש אוכל להתחבר למערכת פעם אחת ומערכת תזכור אותי עד שאתנתק ממנה, על מנת שלא אצטרך לבצע התחברות בכל שימוש באפליקציה. | login\_1 |
| Epic 2- #2 | כמשתמש אוכל לבחור מתוך עמוד ההתחברות במעבר לעמוד ההרשמה על ידי לחיצה על כפתור "הרשמה" על מנת שאוכל לעבור לעמוד ההרשמה במקרה שעדיין אין לי משתמש להתחבר בעזרתו למערכת. | login\_2 |
| Epic 2- #3 | כמשתמש אוכל להזין את השם משתמש שלי (אימייל) ואת הסיסמא על מנת שאוכל להתחבר למערכת. | login\_3 |
| Epic 2 - #4 | כמשתמש אוכל ללחוץ על הכפתור "שכחתי סיסמא" על מנת שאוכל לשחזר את הסיסמא שלי במידה ושכחתי אותה. | login\_4 |
| Epic 2 - #5 | כמשתמש אוכל ללחוץ על כפתור האישור בעמוד התחברות במידה ושם המשתמש והסיסמא נכונים על מנת לבצע את ההתחברות למערכת. | login\_5 |
| Epic 2 - #6 | כמפתח ארצה שיהיה בעמוד ההתחברות כפתור "הרשמה" שיוביל את המשתמש לעמוד ההרשמה במידה ועדיין אין לו משתמש להתחבר בעזרתו כדי שיוכל להירשם למערכת. | login\_6 |
| Epic 2 - #7 | כמפתח ארצה שיהיה בעמוד ההתחברות כפתור "שכחתי סיסמא" על מנת שהמשתמש יוכל לשחזר את הסיסמא שלו במידה ושכח אותה. | login\_7 |
| Epic 2 - #8 | כמפתח ארצה שתופיע הודעת שגיאה במידה והמשתמש הכניס שם משתמש לא תקין, על מנת שיוכל לתקן את זה ולהתחבר למערכת. | login\_8 |
| Epic 2 - #9 | כמפתח ארצה שתופיע הודעת שגיאה במידה והמשתמש הכניס סיסמא לא תקינה, על מנת שיוכל לתקן את זה ולהתחבר למערכת. | login\_9 |
| Epic 2 - #10 | כמפתח ארצה להציג הודעת שגיאה במידה והתיבה של השם משתמש נשארה ריקה , על מנת שהמשתמש ידע ויכניס את שם המשתמש שלו. | login\_10 |
| Epic 2 - #11 | כמפתח ארצה להציג הודעת שגיאה במידה והתיבה של הסיסמא נשארה ריקה , על מנת שהמשתמש ידע ויכניס את שם המשתמש שלו. | login\_11 |
| Epic 2 - #12 | כמפתח ארצה שיהיה בעמוד ההתחברות כפתור אישור שיחבר את המשתמש למערכת לאחר שהכניס את הפרטים הדרושים, על מנת שיוכל להתחבר למערכת. | login\_12 |
| Epic 2 - #13 | כמפתח ארצה שהמשתמש יעבור לעמוד הפתיחה לאחר שהכניס פרטים ולחץ על כפתור האישור התחברות, על מנת שיוכל להתחיל את השימוש במערכת. | login\_13 |
| Epic 2 - #14 | כמפתח ארצה שתהיה תיבת טקסט לשם משתמש ולסיסמא בעמוד ההתחברות, על מנת שהמשתמש יוכל להזין פרטים ולהתחבר למערכת. | login\_14 |
| Epic 2 - #15 | כמפתח ארצה שכאשר המשתמש ילחץ על הכפתור "שכחתי סיסמא" הוא יעבור לעמוד נפרד ושם יזין כתובת מייל לשליחת סיסמא חדשה, על מנת שיוכל לשחזר את הסיסמא שלו. | login\_15 |
| Epic 2 - #16 | כמשתמש אוכל להזין את האימייל שלי בעמוד של "שכחתי סיסמא" על מנת שאוכל לשחזר אותה. | login\_16 |
|  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Designed by:** | | | | **Test Case ID:** | | |
| **Test Designed date:** | | | | **Test Priority** (Low/Medium/High): High | | |
| **Test Executed by:** | | | | **Module Name:** | | |
| **Test Execution date:** | | | | **Test Title:** המערכת, תציג בדף ההקלטה לאחר שלא בוצעה הקלטה ולאחר לחיצה על כפתור ה"איקס" הודעה למשתמש "לא ניתן למחוק הקלטה טרם בוצעה | | |
| **Description:** | | | | | | |
| **Pre-conditions**  : האפליקציה פתוחה ופועלת בדף הקלטה. | | | | | | |
| **Notes** | **Status** (Pass/Fail) | **Actual Result** | **Expected Result** | | **Test Steps** | **Step** |
|  |  | המסך מועבר למילה "מלפפון" | המסך יועבר למילה "מלפפון" | | לחיצה על המילה "מלפפון". | **1** |
|  | pass | מוצג חלון הודעה "לא ניתן למחוק הקלטה טרם בוצעה" | יוצג חלון הודעה "לא ניתן למחוק הקלטה טרם בוצעה" | | לחיצה על כפתור "איקס" | **5** |

**References**

1. <https://he.wikipedia.org/wiki/%D7%9E%D7%A2%D7%A8%D7%9B%D7%AA_%D7%96%D7%99%D7%94%D7%95%D7%99_%D7%93%D7%99%D7%91%D7%95%D7%A8>
2. <https://he.wikipedia.org/wiki/%D7%A4%D7%9C%D7%90%D7%98%D7%A8#%D7%A9%D7%A4%D7%AA_%D7%93%D7%90%D7%A8%D7%98_Dart>
3. <https://he.wikipedia.org/wiki/Dart>
4. <https://www.dogma.co.il/%D7%9E%D7%94-%D7%96%D7%94-%D7%A2%D7%99%D7%91%D7%95%D7%93-%D7%A9%D7%A4%D7%94-%D7%98%D7%91%D7%A2%D7%99%D7%AA-nlp-nlu/>
5. [מערכת זיהוי דיבור – ויקיפדיה (wikipedia.org)](https://he.wikipedia.org/wiki/%D7%9E%D7%A2%D7%A8%D7%9B%D7%AA_%D7%96%D7%99%D7%94%D7%95%D7%99_%D7%93%D7%99%D7%91%D7%95%D7%A8)
6. [למידת מכונה – ויקיפדיה (wikipedia.org)](https://he.wikipedia.org/wiki/%D7%9C%D7%9E%D7%99%D7%93%D7%AA_%D7%9E%D7%9B%D7%95%D7%A0%D7%94#%D7%9E%D7%98%D7%A8%D7%95%D7%AA_%D7%95%D7%A9%D7%99%D7%9E%D7%95%D7%A9%D7%99%D7%9D)
7. [10 Best Voice Recognition Software (Speech Recognition in 2021) (softwaretestinghelp.com)](https://www.softwaretestinghelp.com/voice-recognition-software/)
8. [Buy Dragon Professional Individual - Speech Recognition | Nuance](https://shop.nuance.co.uk/store/nuanceeu/en_GB/Content/pbPage.dragon-professional-individual?currency=EUR&pgmid=95401100&campaignid=1476392048&adgroupid=60832199247&targetid=kwd-329879188924&matchtype=e&device=c&gclid=Cj0KCQjwkbuKBhDRARIsAALysV4QjkDQryaYMDD6LvNkDAazSmm9BzFlgYfI4IxKM8QQ2Fl2G73kX0EaAqN5EALw_wcB&gclsrc=aw.ds)
9. [Google Now – ויקיפדיה (wikipedia.org)](https://he.wikipedia.org/wiki/Google_Now)
10. [סירי – ויקיפדיה (wikipedia.org)](https://he.wikipedia.org/wiki/%D7%A1%D7%99%D7%A8%D7%99)
11. [Amazon Lex – AWS Chatbot AI](https://aws.amazon.com/lex/)
12. [אמזון אלכסה – ויקיפדיה (wikipedia.org)](https://he.wikipedia.org/wiki/%D7%90%D7%9E%D7%96%D7%95%D7%9F_%D7%90%D7%9C%D7%9B%D7%A1%D7%94)
13. <https://www.naya-college.co.il/what-is-machine-learning/>
14. [עיבוד שפה טבעית – ויקיפדיה (wikipedia.org)](https://he.wikipedia.org/wiki/%D7%A2%D7%99%D7%91%D7%95%D7%93_%D7%A9%D7%A4%D7%94_%D7%98%D7%91%D7%A2%D7%99%D7%AA)
15. [מה זה למידת מכונה (Machine Learning)? - BDO](https://www.bdo.co.il/he-il/%D7%91%D7%9C%D7%95%D7%92%D7%99%D7%9D/bdo-tech-hub/%D7%9E%D7%A8%D7%A5-2019/%D7%9C%D7%9E%D7%99%D7%93%D7%AA-%D7%9E%D7%9B%D7%95%D7%A0%D7%94-(machine-learning))
16. <https://he.wikipedia.org/wiki/%D7%91%D7%99%D7%A0%D7%94_%D7%9E%D7%9C%D7%90%D7%9B%D7%95%D7%AA%D7%99%D7%AA>
17. Wagner, S. (2005, May). Intralingual Speech-to-text conversion in real-time: Challenges and Opportunities. In *Challenges of Multidimensional Translation Conference Proceedings*.‏   
    https://www.researchgate.net/profile/Susanne-Wagner-10/publication/283123585\_Intralingual\_speech-to-text-conversion\_in\_real-time\_Challenges\_and\_Opportunities/links/562c029c08aef25a2441ce46/Intralingual-speech-to-text-conversion-in-real-time-Challenges-and-Opportunities.pdf
18. <https://he.wikipedia.org/wiki/%D7%90%D7%A4%D7%9C%D7%99%D7%A7%D7%A6%D7%99%D7%99%D7%AA_%D7%9E%D7%95%D7%91%D7%99%D7%99%D7%9C>
19. Islam, R., Islam, R., & Mazumder, T. (2010). Mobile application and its global impact. *International Journal of Engineering & Technology (IJEST)*, *10*(6), 72-78.‏
20. <https://www.geeksforgeeks.org/difference-between-native-apps-and-web-apps/>
21. [מהי אפליקציה היברידית? יתרונות וחסרונות 2020 (hybridapplication.co.il)](https://www.hybridapplication.co.il/hybrid-application/)
22. <https://www.techtarget.com/searchmobilecomputing/definition/Google-Firebase>
23. [מה זה תמלול ולמה הוא נחוץ? - 100% מקצועיות ואיכות - חבר תרגומים (hevertranslations.co.il)](https://www.hevertranslations.co.il/what-is-transcription/)