



המחלקה להנדסת תוכנה

תיוג ואחסון תמונות

חיבור זה מהווה חלק מהדרישות לקבלת
תואר ראשון בהנדסה

מאת: יונתן ידיד
ת"ז: 304898927

יוני 2018

תמוז, תשע"ח

המחלקה להנדסת תוכנה

תיוג ואחסון תמונות

חיבור זה מהווה חלק מהדרישות לקבלת
תואר ראשון בהנדסה

מאת

יונתן ידיד

(חתימה)

מנחה אקדמי: ד"ר מרים אללוף	אישור:	תאריך:
אחראי תעשייתי: מר אלישי כהן	אישור:	תאריך:
רכז הפרויקטים: ד"ר אסף שפנייר	אישור:	תאריך:

תקציר

כחלק מלימודי הנדסת תוכנה לתואר ראשון בעזריאלי - המכללה האקדמית להנדסה ירושלים, ביצעתי פרויקט גמר במהלך שנת הלימודים הרביעית עבור הספרייה הלאומית של ישראל.

ברשות הספרייה קיים מאגר עצום של תמונות דיגיטליות שונות ומגוונות אשר צולמו במשך עשורים רבים, מלפני קום המדינה ועד היום. התמונות כוללות אתרים, אישים, אירועים ועוד.

הספרייה מעוניינת להנגיש את התמונות לקהל המבקרים שלה. הבעיה מתעוררת כאשר מנסים לחשוב על חיפוש של מבקר במאגר התמונות. אין דרך פשוטה לאתר תמונות בעלות תוכן מסוים מלבד חיפוש ידני עד אשר הן נמצאות.

המטרה העיקרית של הפרויקט הייתה לספק מטא-דאטה למאגר התמונות ולאחסן את המידע בצורה נוחה ויעילה כך שבהמשך ניתן יהיה להקים מנוע חיפוש לתמונות. בנוסף, יצירת מערכת המאפשרת ניהול של המידע השמור, והוספה של מידע חדש במידה והספרייה מוסיפה תמונות חדשות למאגר שלה.

דו"ח זה מתאר את תהליך ביצוע הפרויקט על כל היבטיו ומרכיביו עד לתוצר הסופי, בנוסף לפירוט רלוונטי.



הצהרה:

העבודה נעשתה בהנחיית ד"ר מרים אללוף
במחלקה להנדסת תוכנה, עזריאלי –
המכללה האקדמית להנדסה ירושלים
ובחברת הספרייה הלאומית של ישראל.
החיבור מציג את עבודתי האישית ומהווה
חלק מהדרישות לקבלת תואר ראשון
בהנדסה.

תוכן העניינים

פרק 1 – תיאור מסגרת הפרויקט	6
פרק 2 – תיאור הבעיה	7
פרק 3 – תיאור הפתרון	8
פרק 4 – תיאור המערכת שמומשה	12
פרק 5 – תכנית בדיקות	13
פרק 6 – סקר שוק והשוואה	16
פרק 7 – מסקנות מהפרויקט	17
פרק 8 – ספרות	18
פרק 9 – קישורים למערכות ניהול הפרויקט ובקרת תצורה	19
פרק 10 – נספחים	20

1. תיאור מסגרת הפרויקט

כידוע כיום מערכות מחשבים יכולות לכלול כמויות גדולות של מידע. על מנת לנהל את כל המידע בצורה יעילה, ישנה חשיבות רבה לדרך שבה המידע שמור, ממין ומשותף (כמו כן לדרך שבה נערך חיפוש על המידע).

השיטה הישנה והנפוצה לקבץ ולארגן מידע היא תיקיות. אחד החסרונות המשמעותיים של שיטה זו, במיוחד עבור ארגון תמונות דיגיטליות, הוא שפריט מסוים יכול להימצא רק בתיקייה אחת, אלא אם משכפלים אותו. מצד אחד, שכפול הפריט יהיה בזבוז של מקום בזיכרון וייצור בלבול בניסיון לנהל מעקב אחרי מספר עותקים של אותו פריט. מצד שני, שיוך הפריט רק לתיקייה אחת יצריך החלטה מהי התיקייה המתאימה ביותר.

השיטה החדשה היא תיוג פריטים בתגיות. תגיות הן מילות מפתח אשר משמשות לתאר חתיכת מידע, בין אם מדובר בדף אינטרנט, תמונה דיגיטלית או סוג אחר של מסמך דיגיטלי. שיטה זו מקלה על הדילמה כיצד לקטלג את הפריטים כאשר כל פריט יכול להיות מתויג בתגיות המתאימות לו. הכוח הגדול שבתגיות בא לידי ביטוי כאשר מתבצע חיפוש תמונות. כעת אין צורך לזכור תיקייה ספציפית שבה נמצאת התמונה, אלא לחשוב על תגיות מסוימות אשר עשויות לתאר את התמונה. בחיפוש אחר תגית מסוימת יופיעו כל התמונות שהיא משויכת להן.

בשנים האחרונות הקונספט של תיוג הפך פופולרי. דוגמא בולטת לשימוש כיום היא של חברת "פייסבוק" המאפשרת תיוג אנשים בתמונות. אם מתייגים שמות אנשים לפרצופים בתמונה, אפשר לאתר את התמונות של אדם מסוים במהירות. פירוט נרחב יותר על השימושים כיום ניתן למצוא בפרק 6.

פרויקט זה התמקד במידע מסוג תמונות דיגיטליות. מטרת הפרויקט הייתה ליצור מערכת להכנת תשתית להקמה של מנוע חיפוש תמונות יעיל ואפקטיבי. באופן כללי ניתן לחלק את הפרויקט לשני נושאים מרכזיים: תיוג ואחסון. בכל אחד מהנושאים הייתה עבודה שהתבטאה בדרכי המימוש האפשריות, ובנוסף היה צורך לבצע אינטגרציה בין שני הנושאים כך שבשלב מאוחר יותר ניתן יהיה ליצור מנוע חיפוש למאגר התמונות.

2. תיאור הבעיה

רקע על הארגון ועל המצב של מאגר התמונות שברשותו:

הספרייה הלאומית של ישראל היא הגוף הלאומי מטעם מדינת ישראל המופקד על שמירת אוצרותיה המודפסים של מדינת ישראל, ושל תרבות העם היהודי. בספרייה רשומים מעל לחמישה מיליון פריטי דפוס וארכיון. אלה כוללים גם מאגר עצום של תמונות.

המצב הוא שאין סדר וארגון של מאגר התמונות ברמה שתהיה אפשרות למצוא במהירות תמונות מבוקשות, ובנוסף אין מספיק מידע על כל תמונה. המידע הקיים הוא שמות התמונות, והארגון הוא חלוקה לתתי מאגרים עם תוכן דומה (למשל תמונות של מפות או תמונות שחור לבן). לכן יצירת מנוע חיפוש תמונות לא רלוונטית במצב זה, מאחר ואין דרך לזהות או לאתר תמונות מסוימות במאגר. החיפוש היחיד שניתן אולי לממש במצב הנתון הוא על פי אוסף מאוד מצומצם של מילים המייצגות את תוכן תתי המאגרים. חיפוש כזה הוא בעל רמת דיוק נמוכה (יתקבלו הרבה תמונות עבור מילה כללית), ולא יעיל מבחינת זמן ריצה.

דרישות ואפיון הבעיה

נתת פרק זה מתאר את דרישות הפרויקט מנקודת מבטו של הלקוח, כפי שהציג בתחילת הפרויקט.

תיוג:

יש צורך לספק דרך ממוחשבת אוטומטית לביצוע תיוג התמונות. התיוגים יהיו תיאורי טקסט של תוכן התמונות. רצוי שיהיו אופציות נרחבות ככל שניתן לתיוג התמונות: מעבר להפקת מילות מפתח (תגיות), יכולת לזהות טקסט בתמונות (עבור תמונות כדוגמת מפות) או יכולת לזהות פרצופים בתמונות. התיוג צריך להיות גמיש ככל שניתן: לנסות לתאר את תוכן התמונה על ידי תיאורים שונים שעשויים להיות אפשריים, החל מתיאורים ספציפיים ועד לתיאורים כלליים. עבור כל תיאור רצוי שתהיה אינדיקציה כלשהי מהי מידת ההתאמה שלו לתמונה.

אחסון:

המידע שמתקבל מהתיוגים צריך להיות מאוחסן במסד נתונים כך שעבור כל תמונה תהיה אפשרות לגשת לתיוגים השונים שבוצעו לה (לכל תמונה יהיה מזהה ייחודי). עבור כל תמונה עשוי להיות תיוג המתבטא בגודל זיכרון שונה ולכן יש צורך במסד נתונים שיתמוך באחסון תיוגים בעלי מבנה וגודל שונים. יש צורך בגישה פשוטה לאחסון לצורך הוספה/מחיקת מידע. האחסון צריך להיות יעיל ככל שניתן על מנת שתהיה אפשרות לחפש תמונות לפי מילת מפתח.

3. תיאור הפתרון

מערכת ייחודית בעלת ממשק משתמש פשוט ונוח שלא מצריך ידע טכני מיוחד, כך שכל בעל תפקיד מורשה יכול להשתמש במערכת. המערכת זמינה בכל רגע נתון ומופעלת על מחשב רצוי בהפעלה מהירה ופשוטה. המערכת מספקת למשתמש מגוון אפשרויות לזיהוי תוכן תמונות: זיהוי אובייקטים, זיהוי סמלילים (לוגואים), זיהוי טקסט ומיקומו בתמונה, זיהוי פרצופים ומיקומם בתמונה, זיהוי ישויות מוכרות או זיהוי אתרים מוכרים. המערכת אחראית גם על אחסון והיא מאפשרת למשתמש לשמור את התוכן שהיא זיהתה בתמונות, או למחוק/לשנות תוכן שמור בכל עת. המערכת מספקת אפשרויות חיפוש על מאגר תמונות שעבר תיוג ואחסון: ניתן לחפש תמונות בכמה אפשרויות שונות, בין השאר על ידי מילת/מילות תיאור התמונות.

ארכיטקטורת המערכת:

השיקולים שנלקחו בחשבון בבניית הארכיטקטורה הם מודולריות של הקוד והפרדה לתפקידים. לכל מודול צריך להיות תפקיד מוגדר ללא העמסה של קוד מיותר, בניסיון לבנות את המערכת בצורה גמישה שמאפשרת עדכון קוד קל במידת הצורך. בניית הממשק הגרפי נעשתה בתכנות מונחה עצמים כך שכל אובייקט מייצג עמוד שונה עם רכיבים גרפיים שונים ואפשרויות שונות.

המערכת מורכבת ממספר מודולים:

tagging.py – המודול האחראי על עיבוד תמונה.
הוא עושה שימוש ב-Google Vision API ובעת שימוש הוא יוצר חבילת בקשה עבור תמונה מסוימת אשר מכילה מידע נדרש, ומחזיר את התוצאות שהתקבלו לאחר עיבוד התמונה.

storage.py – המודול האחראי על קישור למסד הנתונים.
הוא מספק את כל השיטות אשר עושות שימוש במסד הנתונים כגון אחסון או מחיקת מידע.
בנוסף הוא מספק שיטות חיפוש שונות במסד הנתונים.

image_options.py – המודול מספק מניפולציות שונות על קובצי תמונות במחשב כמו שיטה לסימון או חיתוך פרצופים ושמירת התמונות החדשות במחשב.

alpha.py – תפריט טרמינלי אינטראקטיבי המאפשר למשתמש לבצע את מגוון הפעולות הקיים לרשותו, החל מתיוג ואחסון תמונות, ועד לחיפושים במסד הנתונים.

beta.py – אובייקט המייצג את העמוד העיקרי בממשק משתמש גרפי לגרסת האלפא (alpha.py).
על גבי האובייקט הזה ייוספויימחקו שאר האובייקטים לפי הצורך.

beta_tag_store.py – אובייקט המייצג את הרכיבים הגרפיים והפונקציונליות של העמוד לתיוג
ואחסון תמונה.

beta_storage.py – אובייקט המייצג את הרכיבים הגרפיים והפונקציונליות של העמוד לגישה
ושינוי האחסון.

beta_search_content.py – אובייקט המייצג את הרכיבים הגרפיים והפונקציונליות של העמוד
לחיפוש תוכן של תמונה.

beta_search_images.py – אובייקט המייצג את הרכיבים הגרפיים והפונקציונליות של העמוד
לחיפוש תמונות לפי מילות מפתח.

תרשים מס' 1 בפרק 10 מתאר את ארכיטקטורת המערכת.

תהליכים ופעולות המערכת:

המערכת כוללת תהליך אחד בכל זמן נתון, ועל גבי תהליך זה מתבצעות מגוון הפעולות. בכל בחירה
של המשתמש לביצוע פעולה כלשהי, התהליך מבצע את הפעולה והמערכת ממתינה לסיום הפעולה,
כאשר בסיום הפעולה המערכת מתפנה לביצוע פעולה נוספת.

תרשים מס' 2 בפרק 10 מתאר את אינטראקציית המשתמש עם המערכת.

אופן ביצוע עיבוד תמונה וזיהוי תגיות:

המערכת מקבלת קובצי תמונות מהמאגר ועושה שימוש בשיטה images.annotate שמסופקת על ידי
Google Vision API. הפעלת השיטה images.annotate מבוססת על REST API. על המערכת לשלוח
בקשת HTTP POST לכתובת <https://vision.googleapis.com/v1/images:annotate>.
הבקשה תכלול פרמטר הנקרא key, וגוף הבקשה יכיל מידע על התמונה הרצויה והגדרות נוספות
שלפיהן תפעל השיטה, בין היתר סוג עיבוד התמונה המבוקש. הפרמטר key הוא מפתח פרטי אותו
יש לקבל מראש מ"גוגל", והוא מאפשר אותנטיקציה של הלקוח המעוניין להשתמש בממשק.

בגוף התגובה שמתקבלת חזרה יופיעו תוצאות מהרצת השיטה. מבנה גוף התגובה והתוצאות המתקבלות לאחר עיבוד התמונה, תלויים בסוג עיבוד התמונה הרצוי. הן גוף התגובה והן גוף ההודעה יהיו בפורמט JSON כמוחלט לפי הממשק.

טבלה מס' 1 בפרק 10 מתארת את מבנה חבילת התגובה לכל סוג בקשה.

עלויות ומגבלות של שימוש בממשק:

- יש עלות כספית לשימוש בממשק והיא מחושבת בהתאם להיקף השימוש בחודש.
 - הממשק תומך במספר פורמטים מסוימים של תמונות.
 - הממשק תומך בגדלי קבצים בטווח מסוים.
- ניתן לראות פירוט לגבי כל אחד מהסעיפים בקישורים המופיעים בפרק 8.

דרכי ההתמודדות:

- הלקוח מודע לנושא העלות ויש מעקב אחר היקף השימוש.
- לאחר שחקרתי את מגבלות הפורמט והגודל מתברר שקיימות האפשרויות להמיר פורמט תמונה שלא נתמך ע"י הממשק לפורמט נתמך, וכן לדחוס גודל תמונה שלא נתמך לגודל נתמך. אולם במערכת הנתונה אין צורך לטפל במגבלות אלו מאחר ולפי דברי הלקוח, מאגר התמונות כולו עומד במגבלות.

אופן ביצוע אחסון:

האחסון הוא מסד נתונים לא רלציוני. הוא מותקן על שרת/שרתים המאזינים לבקשות HTTP על פורט מסוים. המערכת יוצרת חיבור עם השרת/שרתים ומתקשרת עם מסד הנתונים באמצעות REST API. החיבור נעשה בתחילת ההפעלה של המערכת והוא ממשיך כל עוד המערכת ממשיכה לפעול.

עבור כל אפשרות זיהוי (פרצופים, אובייקטים וכו') מוגדר אינדקס: זהו למעשה ה"כותרת" של הנתונים השמורים תחתיו. לכל תמונה שנשמרת יש מזהה ייחודי שהוא שם התמונה כולל הסיימת שלה, ותחת המזהה הייחודי שמור המידע על התמונה שמיוצג על ידי מחרוזת JSON.

המערכת שולחת בקשת HTTP PUT עם המזהה הייחודי של התמונה, האינדקס הרצוי לשמירה, והמידע עצמו כמחרוזת JSON. במסד הנתונים תחת האינדקס הרצוי נשמרים מזהה התמונה עם המידע.

תרשים מס' 3 בפרק 10 מתאר את אופן ביצוע תיוג ואחסון תמונה.

אופן ביצוע שליפה/חיפוש במסד הנתונים :

במקרה של שליפת מידע על תמונה ספציפית יש צורך לשלוח למסד הנתונים חבילה שמכילה את האינדקס שבו המידע שמור ואת המזהה הייחודי של התמונה. החבילה החוזרת תכיל את המידע השמור של אותו מזהה ייחודי בתוך אותו אינדקס.

במקרה של חיפוש תמונות המכילות מידע משותף כגון תווית זהה או טקסט זהה, יש צורך להוסיף לחבילת השליחה שאילתה כתובה בשפה Query DSL – השפה המקובלת להפעלת שאילתות במסד הנתונים שבחרתי לפרויקט זה. השאילתה מופעלת ומחפשת התאמות בכל האינדקסים הקיימים במסד הנתונים. החבילה החוזרת תכיל את כל התמונות (מיוצגות על ידי המזהה הייחודי שלהן) שהמידע שלהן תואם לשאילתת החיפוש. מידע על שפת השאילתות מופיע בקישור בפרק 8.

* כל הפעולות (עדכון/מחיקה/שליפה/חיפוש) במסד הנתונים וכל הפעולות לעיבוד תמונה מתבצעות על ידי REST API. טבלה מס' 2 בפרק 10 מתארת את חבילות HTTP הנדרשות לשימושים שונים במסד הנתונים ובממשק עיבוד תמונה.

תיאור הכלים ששימשו לפתרון

בפרויקט השתמשתי במספר כלים לפתרון :

Google Vision API : ממשק עיבוד תמונה אשר מסופק על ידי חברת "גוגל", המאפשר למפתחים להעביר למערכת תמונות ולקבל חזרה סט מידע על התמונות הכולל נתונים כמו צבעים, זיהוי פנים והבעות פנים, זיהוי אובייקטים ומיקומים בעולם, זיהוי לוגואים ועוד.

Elasticsearch : מנוע חיפוש המבוסס על Lucene (ספריית תוכנה לאיתור ואיסוף מידע). הוא מספק, בין היתר, אינספור אופטימיזציות על מנת לספק ביצועים מהירים מאוד ומאפשר לבצע אנאליזות על המסמכים, סכימה גמישה והתקנה מבוזרת. זהו מנוע חיפוש עוצמתי וגמיש שמסוגל להתמודד עם כמויות מידע עצומות.

מחשב בעל מערכת הפעלה Windows 10 עם חיבור אינטרנט.

כתיבת הקוד נעשתה בשפת פייתון (Python) גרסה 3.6.3.

4. תיאור המערכת שמומשה

תוכנה בעלת ממשק גרפי פשוט, נקי, וקל לתפעול. התוכנה כוללת מספר עמודים שלכל אחד מהם יש ייעוד מסוים ומאפשרת לנווט בין העמודים.

עמוד לתיג ואחסון תמונה :

בעמוד זה המשתמש יכול לבחור תמונה מהזיכרון ולהציג אותה על התוכנה. יש אפשרות להשתמש בטכניקת עיבוד תמונה מתוך 6 אפשרויות זיהוי (אובייקטים, פרצופים, לוגואים, תמונות דומות/זרות באינטרנט, אתרים וטקסט). במקרה של זיהוי פרצופים, התמונה תוצג בתוכנה בתוספת סימון של הפרצופים. את התוצאות מכל סוג עיבוד אפשר לבחור לאחסן.

עמוד לגישה ושינוי האחסון :

בעמוד זה המשתמש יכול לקבל מידע אודות כל התמונות המידע עליהן שמור באחסון. העמוד מדמה למעשה את המבנה של האחסון בצורה פשוטה וידידותית. בנוסף אפשר למחוק מידע על תמונות רצויות מכל אינדקס באחסון.

עמוד לחיפוש תוכן של תמונה :

העמוד מאפשר חיפוש מהיר של תוכן באחסון לפי מזהה ייחודי של תמונה (שם התמונה). בכל אחד מהאינדקסים תתקבל תשובה האם מאוחסן מידע על התמונה המבוקשת, ואם כן, יש אפשרות לראות את המידע.

עמוד לחיפוש תמונות לפי מילות מפתח :

העמוד מאפשר למשתמש להזין מילות מפתח ולקבל את כל התמונות שמידע עליהן באחסון כולל את מילות המפתח. התמונות יופיעו בתוכנה במידה והן קיימות בנתיב שהן התקבלו ממנו לצורך העיבוד, וניתן לראות אותן בגודל מקורי ולקבל את השם והנתיב. במידה והתמונה לא קיימת בנתיב, יתקבל מידע אודות השם והנתיב האחרון, ללא הצגה של התמונה. העמוד למעשה מדמה מנוע חיפוש לתמונות שכבר עברו זיהוי ויש עליהן מידע שמור.

5. תכנית בדיקות

כאמור המערכת מורכבת ממספר מודולים האחראים על תפקידים שונים. הבדיקות הראויות לביצוע בפרויקט זה הן:

- בדיקות יחידה – בדיקות עבור הפונקציות השונות במודולים, על מנת להבטיח שהפונקציות עושות את מה שהן אמורות לעשות, ולא עושות את מה שהן לא אמורות לעשות.
- בדיקות אינטגרציה – בדיקות הכוללות את תקינות ונכונות הפונקציות של המודולים השונים בשימוש במודולים אחרים.
- בדיקות מסד נתונים – בדיקות הכוללות את תקינות ונכונות השימוש במסד הנתונים.
- בדיקות ממשק לקוח (GUI) – בדיקות חווית הממשק הגרפי ותפקודו.

סוג הבדיקה	תיאור הבדיקה	תוצאות הבדיקה
יחידה	הפעלת השיטה להוספת תוכן של תמונה עם תמונה שלא קיימת על המחשב.	השיטה תופסת חריגה ומודיעה על שגיאה בניסיון לפתוח את התמונה.
יחידה	הפעלת השיטה להוספת תוכן של תמונה עם תמונה שקיימת על המחשב.	השיטה מחזירה את תוכן התמונה מקודד בבסיס 64, לפי דרישת ממשק העיבוד.
יחידה	הפעלת השיטה להוספת סוג עיבוד שלא קיים במימוש.	השיטה מודיעה על שגיאה בהוספה.
יחידה	הפעלת השיטה להוספת סוג עיבוד שקיים במימוש.	השיטה מחזירה את סוג העיבוד הרצוי לפי פורמט הממשק.
יחידה	הפעלת השיטה לשליחת בקשת עיבוד עם מפתח שלא קיים.	השיטה תופסת חריגה ומודיעה על שגיאה בקבלת מפתח.
יחידה	הפעלת השיטה לשליחת בקשת עיבוד ללא חיבור לאינטרנט.	השיטה תופסת חריגה ומודיעה על שגיאה בשליחת הבקשה.
יחידה	הפעלת השיטה לשליחת בקשת עיבוד עם תמונה לא נתמכת.	השיטה מחזירה ערך False שמעיד על היעדר תוצאות עיבוד.
יחידה	הפעלת השיטה לשליחת בקשת עיבוד עם מפתח שגוי.	השיטה מחזירה ערך False שמעיד על היעדר תוצאות עיבוד.

השיטה מחזירה ערך True שמעיד על תוצאות תקינות ואת תוצאות העיבוד בפורמט JSON	הפעלת השיטה לשליחת בקשת עיבוד עם פרמטרים תקינים.	יחידה
השיטה תופסת חריגה ומודיעה על שגיאה בשמירה.	הפעלת שיטה לסימון פרצופים ושמירה, עם נתיב שמירה שגוי.	יחידה
השיטה תופסת חריגה ומודיעה על שגיאה בשמירה.	הפעלת שיטה לחיתוך פרצופים ושמירה, עם נתיב שמירה שגוי.	יחידה
השיטה מסמנת את הפרצופים ושומרת תמונה חדשה.	הפעלת שיטה לסימון פרצופים ושמירה.	יחידה
השיטה חותכת את הפרצופים ושומרת תמונה חדשה.	הפעלת שיטה לחיתוך פרצופים ושמירה.	יחידה
הנתונים נשמרים בפורמט JSON המבוקש, הנתונים מוחזרים בפורמט JSON המבוקש, הנתונים נמחקים, ומתקבל ערך False בניסיון לשלוח שוב.	שמירת נתונים, שליפת נתונים, מחיקת נתונים, וניסיון שליפה חוזר מאינדקס קיים.	מסד נתונים
אינדקס חדש נוצר והנתונים נשמרים.	שמירת נתונים לפני יצירת אינדקס חדש.	מסד נתונים
מתקבלת שגיאה שהאינדקס לא קיים והפעולות לא מתבצעות.	ניסיון שליפה אחיפוש אחיקה מאינדקס לא קיים.	מסד נתונים
מתקבלות תוצאות חיפוש מכל האינדקסים הקיימים, בפורמט JSON הרצוי.	הפעלת שאילתת חיפוש לפי מילות מפתח ומידת התאמה.	מסד נתונים
השיטה מפעילה את השיטה לבקשת עיבוד עם הפרמטרים המסומנים בממשק הגרפי (כל הפרמטרים תקינים), ומציגה את התוצאות על המסך.	לחיצה על לחצן Process image בעמוד התיוג בממשק הגרפי, לאחר שנבחרה תמונה מהמחשב.	אינטגרציה
השיטה משתמשת בשיטה ממודול האחסון ומקבלת אישור לאחר השמירה.	לחיצה על שמירת נתונים לאחר עיבוד מבוקש.	אינטגרציה
מתקבלות תוצאות לפי האינדקס הרצוי והתוצאות מודפסות על המסך תחת האינדקס הרצוי.	טעינת תמונות ממסד הנתונים בעמוד האחסון.	אינטגרציה

התמונות המסומנות נמחקות מהאינדקס הרצוי ולאחר מכן נמחקות מהמסד.	מחיקת תמונות ממסד הנתונים בעמוד האחסון.	אינטגרציה
המערכת מבצעת חיפוש במסד הנתונים ומדפיסה על המסך את האינדקסים שבהם נמצאה התמונה, כולל לחצן המכיל את התוכן.	חיפוש תוכן של תמונה בעמוד חיפוש תוכן.	אינטגרציה
במערכת מפעילה שאילתה במסד הנתונים ומציגה על המסך את כל התמונות שהתקבלו.	חיפוש תמונות לפי מילות מפתח בעמוד חיפוש תמונות.	אינטגרציה
לחצן עיבוד תמונה מנוטרל כל עוד לא נבחרה תמונה, לחצן שמירת נתונים מנוטרל כל עוד לא התקבלו תוצאות או לאחר שמירה, אפשרות זיהוי מסומנת לאחר בחירה, כל אחד מהלחצנים מגיב ללחיצה ומפעיל שיטה בהתאם.	בדיקת לוגיקה והתנהגות תקינה של הרכיבים בעמוד התיוג והאחסון.	ממשק לקוח (GUI)
לחצנים של טעינה/מחיקה מחוברים לאינדקס רצוי, אין אפשרות למחוק פריטים לא מסומנים, אין אפשרות לסמן בו-זמנית פריטים מאינדקסים שונים, כל אחד מהלחצנים מגיב ללחיצה ומפעיל שיטה בהתאם.	בדיקת לוגיקה והתנהגות תקינה של הרכיבים בעמוד האחסון.	ממשק לקוח (GUI)
בחירת/ביטול אינדקסים לחיפוש מוצגת על המסך ישירות, הטקסט לדוגמא נמחק משורת החיפוש לאחר קבלת פוקוס בפעם הראשונה, תוצאות חיפוש מוצגות בתחום של האינדקס, כל אחד מהלחצנים מגיב ללחיצה ומפעיל שיטה בהתאם, כל אחד מהלחצנים מגיב למעבר הסמן עליו ומשתנה בהתאם.	בדיקת לוגיקה והתנהגות תקינה של הרכיבים בעמוד חיפוש תוכן לפי תמונה.	ממשק לקוח (GUI)

ממשק לקוח (GUI)	בדיקת לוגיקה והתנהגות תקינה של הרכיבים בעמוד חיפוש תמונות.	לחצן לטעינת תמונות נוספות מוצגנמחק מהמסך בהתאם לצורך, לחצן פירוט תמונות שלא נמצאות בזיכרון המחשב מוצגנמחק מהמסך בהתאם לצורך, חיפוש חדש מוחק תוצאות חיפוש קודם, סרגל גלילה מתאפשר כשמתווספות תוצאות בתחתית העמוד ומתעדכן בהתאם, כל הרכיבים מגיבים לאירועים ומפעילים שיטה בהתאם.
ממשק לקוח (GUI)	מעברים בין עמודים שונים, כולל אחרי פעילויות שונות בעמודים	העמוד הקודם נמחק והעמוד החדש נטען, כולל רכיבים ופונקציונליות תקינים.

6. סקר שוק והשוואה

בפרק זה אדון במוצרים והשירותים הקיימים בשוק, וברמת הדמיון והרלוונטיות שלהם לפרויקט.

אפליקציית גוגל תמונות: האפליקציה משמשת כגלריית תמונות, והיא מספקת אפשרויות עריכה שונות של תמונות, אנימציות, אלבומים ועוד. בנוסף יש גיבוי של התמונות בענן וארגון אוטומטי של התמונות. דמיון לפרויקט: התמונות באפליקציה ניתנות למציאה על ידי חיפוש אנשים, מקומות או אובייקטים המופיעים בהן ללא צורך בתיוג פיזי. מידת התאמה לפרויקט: תיוג התמונות אוטומטי ואינו ניתן לשליטה מלאה ולהתאמה אישית, אין אפשרות לדעת או לבחור את התגיות של כל תמונה. גם הגיבוי בענן הוא אוטומטי ואינו ניתן להתאמה אישית. לכן אין אפשרות להשתמש באפליקציה זו לצורך הפרויקט, לא בבחירת תגיות ולא בבניית מסד נתונים.

חברות מיקור המונים (Crowdsourcing): חברות אשר משתמשות ב"כוח הקהל" לביצוע עבודות במגוון תחומים. מיקור המונים הוא הפניה של ביצוע משימה או משימות אשר לרוב היו מתבצעות בידי עובדי חברה או ארגון, לביצוע על ידי קהל גדול. חברת מיקור המונים יכולה להציע משימה לקהל הרחב ולהעניק תשלום למבצע המשימה הטוב ביותר, או לחלק את המשימה ואת התשלום בין מספר מבצעים שונים. ההתפתחות הטכנולוגית מאפשרת גישה במקביל למספר גדול של אנשים באמצעות רשת האינטרנט.

כך למשל החברה "Clickworker" מקבלת על עצמה פרויקטים של תיוג תמונות. החברה מקבלת מהלקוח מפרט דרישות הכולל בין השאר את כמות התמונות שהוא מעוניין לתייג, את שפת התיוג ואת כמות התיוגים לתמונה. החברה מחלקת את הפרויקט לעבודות קטנות ומפרסמת אותן לקהל הפריילנסרים המתאים שלה. מידת התאמה לפרויקט: ניתן לפנות לחברה לצורך ביצוע תיוג התמונות, אך מדובר בבניה אנושית ותיוג פיזי של התמונות על ידי אנשים שונים, כך שאין בהכרח תיאום בשמות התגיות. כמו כן, בהוספה של תמונות חדשות למאגר הקיים יהיה צורך בפניה נוספת לחברה. בנוסף אין פתרון לאחסון המידע.

תוכנות לארגון תמונות: קיימות כיום תוכנות כדוגמת ACDSee 20 או Zoner Photo Studio שמאפשרות ארגון יעיל יותר של התמונות בין השאר על ידי הוספת תגיות או מיקומים. מידת התאמה לפרויקט: שוב מדובר בתיוג פיזי ולא בדרך תכנותית. התוכנות פועלות על קבצים המאוחסנים בזיכרון המקומי או בזיכרון נייד, כלומר יש צורך לייבא את התמונות כעבודה מקדימה.

לסיכום, המוצרים הקיימים כיום אכן מביאים לידי שימוש את הקונספט של תיוג אך אינם מהווים פתרון מספק לדרישה זו. נקודה חשובה נוספת היא נושא האחסון: המוצרים הקיימים אינם מהווים פתרון בנושא הביג דאטה. הם אינם בנויים לאחסון ולניתוח כמויות מידע גדולות, ולעמידה באתגרים של כתיבה ושליפה מהירות מאוד. לאחר סקירה מקיפה ניתן לומר שאין מוצר העונה על דרישות הפרויקט במלואן. הפרויקט ייחודי בדרישות שלו ודורש גמישות רבה, התאמה אישית ושליטה מלאה בהתנהלות.

7. מסקנות מהפרויקט

במהלך העבודה על הפרויקט למדתי הרבה הן בתחום הטכני והמקצועי, והן בתחום הניהולי והארגוני. התנסיתי בלימוד עצמאי וביישום של נושאים לא מוכרים כמו שפת תכנות חדשה, ממשק פיתוח ומסד נתונים לא מוכרים. למדתי איך להשתמש בכלים קיימים שרכשתי בלימודים ואיך לשלב עם כלים חדשים. נוכחתי להבין באופן מעשי שתכנון מוקדם, ניהול זמן וניהול משימות הם נתאים הכרחיים להצלחה של פרויקט. למדתי איך להתאים עבודה ומימוש לדרישות ותיאור של לקוח.

באשר לנושא הפרויקט עצמו, גיליתי שלמידע על מידע (מטה-דאטה) יש יתרונות גדולים בארגון של קבצים ובחיפוש אחר קבצים, במיוחד בתקופה המודרנית שכמויות מידע ושטחי אחסון הולכים וגדלים. באופן טבעי במהלך הפרויקט הקונספט של מטה-דאטה תפס את תשומת ליבי ונחשפתי לשימושים דומים כמו אפליקציות ואפילו מערכות הפעלה שמביאות לידי ביטוי את הנושא (למשל מערכת ההפעלה הנוכחית של אנדרואיד מאפשרת לתייג תמונות באופן ידני מתוך הגלריה של הסמארטפון). לדעתי, נושא המטה-דאטה ימשיך להתפתח ויהיה חלק אינטגרלי במערכות ממוחשבות.

8. ספרות

- Python 3.6.3 וספריות חשובות

<https://www.python.org/downloads/release/python-363>
<https://docs.python.org/3>
<http://docs.python-requests.org/en/master>
<https://docs.python.org/3/library/json.html>
<https://docs.python.org/3/library/io.html>
<https://docs.python.org/3/library/sys.html>
<https://docs.python.org/3/library/base64.html>
<https://pillow.readthedocs.io/en/latest>

- Google Vision API

<https://cloud.google.com/vision>
<https://cloud.google.com/vision/docs>
<https://cloud.google.com/vision/docs/reference/rest/v1/images/annotate>
<https://cloud.google.com/vision/docs/request>
<https://cloud.google.com/vision/pricing>
<https://cloud.google.com/vision/docs/supported-files>

Elasticsearch -

[/https://www.elastic.co](https://www.elastic.co)

<https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>

[/http://elasticsearch-py.readthedocs.io/en/master](http://elasticsearch-py.readthedocs.io/en/master)

9. קישורים למערכות ניהול הפרויקט ובקרת תצורה

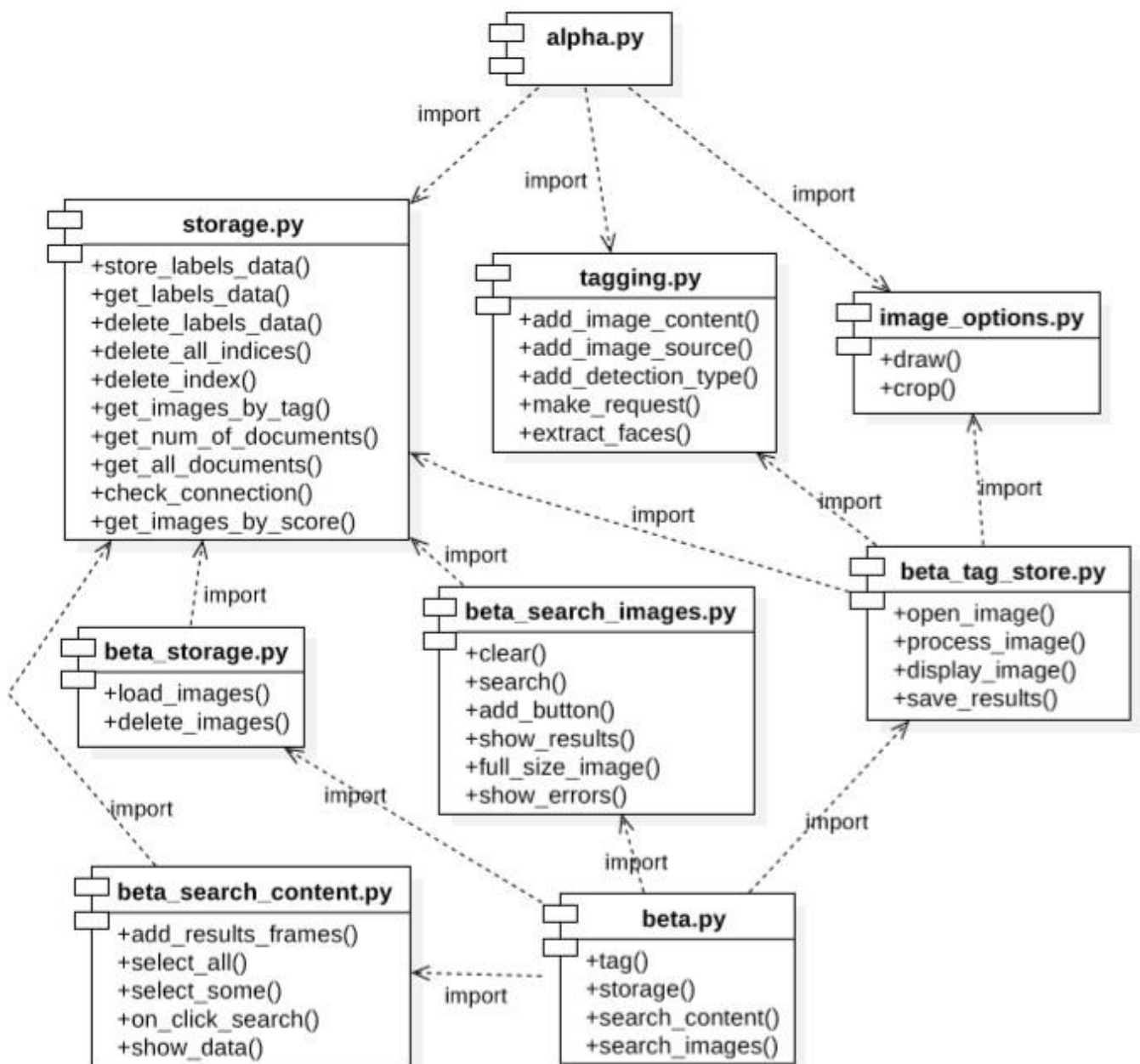
מיקום	מערכת
https://github.com/yonatan-y/pictures-tagging	מאגר קוד
https://github.com/yonatan-y/pictures-tagging/wiki/Project-Diary	יומן
https://github.com/yonatan-y/pictures-tagging/issues	ניהול פרויקט
https://1drv.ms/v/s!AqMm8YLp0zwrhzMrFjRNMactcSJk	סרטון

10. נספחים

תרשימים

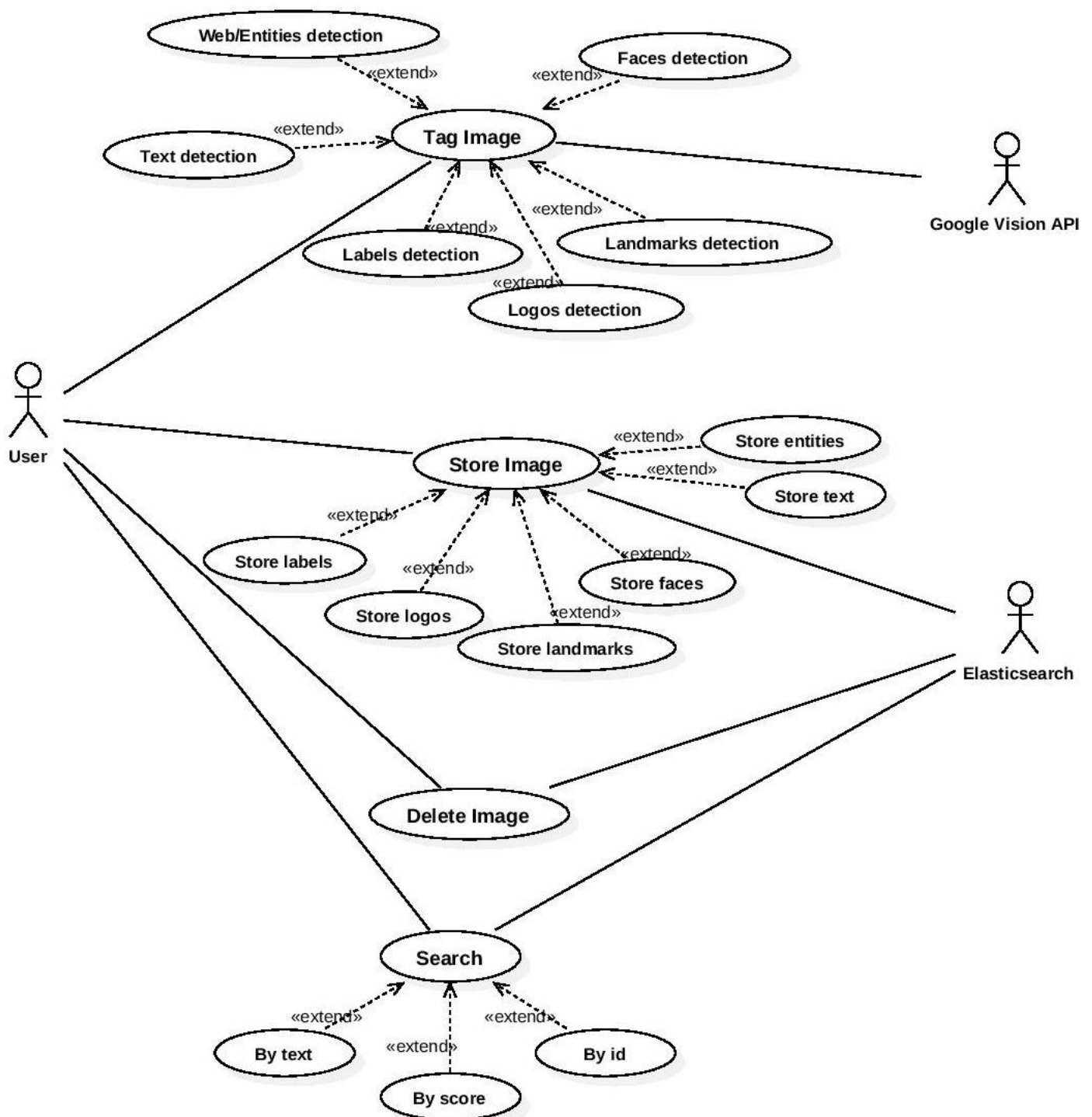
תרשים מס' 1 – ארכיטקטורת המערכת

התרשים מתאר את מבנה הקבצים של המערכת ואת השימושים שלהם בין המערכת. בתרשים מתוארות גם השיטות הממומשות בקבצים (רק שם השיטה ללא פרמטרים או ערכים חוזרים).



תרשים מס' 2 – אינטראקציית המשתמש עם המערכת

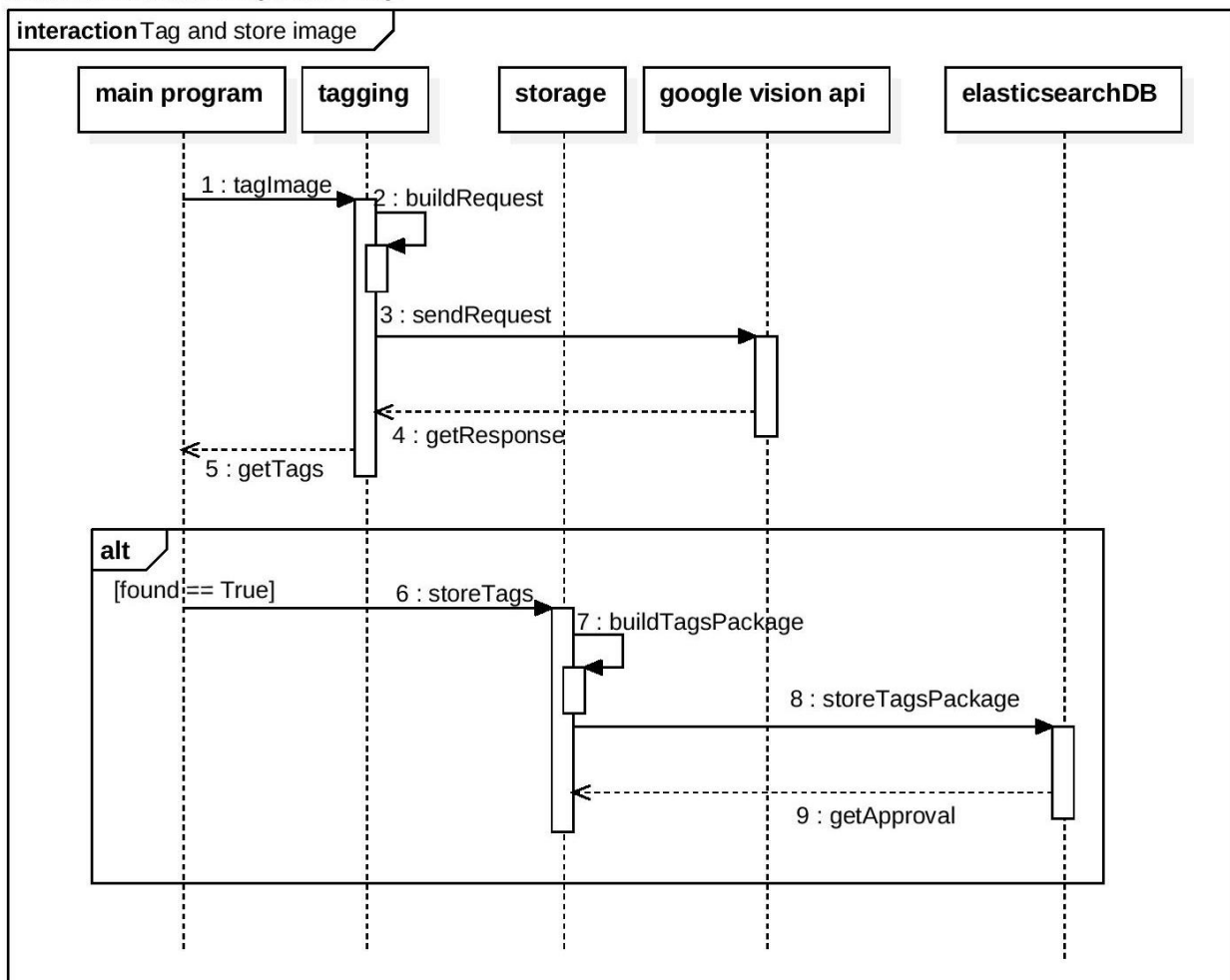
התרשים מתאר את הפעולות שהמשתמש יכול לבצע ואת הגורמים החיצוניים שמעורבים בכל סוג פעולה.



תרשים מס' 3 – תיוג ואחסון תמונה

תרשים רצף המתאר את השלבים בתיוג ואחסון תמונה: המערכת בונה חבילה לבקשת עיבוד תמונה ושולחת את החבילה לממשק העיבוד. ממשיק העיבוד מחזיר תשובה, ובמידה ובגוף התשובה נמצאו זיהויים המערכת יכולה לאחסן את המידע במסד הנתונים ולקבל אסמכתא שהמידע נשמר.

Collaboration1::Interaction1::Tag and store image



טבלאות

טבלה מס' 1 – סוגי בקשות ותגובות

הטבלה הבאה מתארת את מבנה חבילת התגובה לכל סוג בקשה של עיבוד תמונה. כאמור החבילות הן בפורמט JSON. הערה: צד ימין של הטבלה מתאר רק את סוג עיבוד התמונה הרצוי, ולא את מבנה הבקשה הכולל. הסבר ופירוט על כל שדה בחבילות הבקשה והתגובה ניתן למצוא בקישור בפרק 8.

מבנה חבילת תגובה	סוג בקשה
<pre>"labelAnnotations": [{ "mid": string, "description": string, "score": number }]</pre>	LABEL_DETECTION
<pre>"logoAnnotations": [{ "mid": string, "description": string, "score": number, "boundingPoly": { object (BoundingPoly) } }]</pre>	LOGO_DETECTION
<pre>"landmarkAnnotations": [{ "mid": string, "description": string, "score": number, "boundingPoly": { object (BoundingPoly) }, "locations": [{</pre>	LANDMARK_DETECTION

<pre> object (LocationInfo) }] }]</pre>	
<pre> "faceAnnotations": [{ "boundingPoly": { object (BoundingPoly) }, "fdBoundingPoly": { object (BoundingPoly) }, "landmarks": [{ object (Landmark) }], "rollAngle": number, "panAngle": number, "tiltAngle": number, "detectionConfidence": number, "landmarkingConfidence": number, "joyLikelihood": enum(Likelihood), "sorrowLikelihood": enum(Likelihood), "angerLikelihood": enum(Likelihood), "surpriseLikelihood": enum(Likelihood), "underExposedLikelihood": enum(Likelihood), "blurredLikelihood": enum(Likelihood), "headwearLikelihood": enum(Likelihood) }]</pre>	FACE_DETECTION
<pre> "webDetection": { "webEntities": [{ object (WebEntity) }] }</pre>	WEB_DETECTION

<pre>], "fullMatchingImages": [{ object(WebImage) }], "partialMatchingImages": [{ object(WebImage) }], "pagesWithMatchingImages": [{ object(WebPage) }], "visuallySimilarImages": [{ object(WebImage) }] } </pre>	
<pre> "textAnnotations": [{ "locale": string, "description": string, "boundingPoly": { object(BoundingPoly) } }] "fullTextAnnotation": { "pages": [{ "property": { "detectedLanguages": [{ object(DetectedLanguage) }] } }] } </pre>	<p>TEXT_DETECTION</p>

```

    ],
    "detectedBreak": {
      object (DetectedBreak)
    }
  },
  "width": number,
  "height": number,
  "blocks": [
    {
      object (Block)
    }
  ]
}
],
"text": string
}

```

טבלה מס' 2 - סוגי חבילות:

הטבלה הבאה מתארת את סוגי החבילות השונים עבור פעולות שונות של המערכת.

סוג חבילה	פעולה רצויה
HTTP POST	זיהוי תוכן (עיבוד) תמונה
HTTP PUT	אחסון במסד הנתונים
HTTP DELETE	מחיקה ממסד הנתונים
HTTP POST	עדכון במסד הנתונים
HTTP GET	חיפוש/שליפה ממסד הנתונים

Abstract

As part of the studies for B.Sc. in software engineering at Azrieli – college of engineering Jerusalem, during the fourth year of studies I performed a final project for THE NATIONAL LIBRARY OF ISRAEL.

The national library possesses a tremendous repository of different and diverse digital pictures, which has been photographed for many decades from before the establishment of the state until today. The digital pictures include sites, personalities, events, etc.

The library is interested to provide accessibility of the pictures to the visitors. The problem arises when we try to think about a random visitor's search in the pictures repository. There is no simple way to identify pictures with a specific content except manual search until finding.

The main purpose of the project was to provide metadata for the pictures repository and to store the information in an easy and effective way so that it would be possible to create a search engine for the pictures. Also, creating a system that allows management of the stored data, including addition of new data in case that the library receives new pictures.

This report describes the project process including all aspects and components until the final product, in addition to relevant information.



Software Engineering Department

Pictures Tagging And Storing

by

Yonatan Yadid

Academic Supervisor: Dr. Miriam Allalouf



Software Engineering Department

Pictures Tagging And Storing

by: Yonatan Yadid

id: 304898927

June 2018

Tamuz 5778