מסמך אפיון Face Founder במקצוע הגנת סייבר

מגיש : אלמוג גל

נושא הפרוייקט : מציאת פנים.

הגדרת הפרוייקט (המוצר): הפרוייקט Face Founder נועד למציאת פני אדם אחד\רבים ספציפי\ם מתוך מקבץ מידע חזותי. לכל משתמש באפליקציה יהיה user משלו שבו ישמרו נתוניו, כמו היסטוריית חיפוש, בצורה מאובטחת. על מנת להיות בטוח שתמונה מסויימת לא נאבדה בחיפוש, את התמונות שבהם נמצא בין 75-90 (אולי 80-90) אחוז התאמה, אני אשאל את המשתמש האם לדעתו האדם בתמונה הוא האדם שאותו חיפש. בנוסף אתן למשתמש אופציה לראות את אחוזי ההתאמה לכל תמונה שהאדם המבוקש נמצא בה (כך שבמקרה של מציאת אדם אחר לאדם המבוקש המשתמש יוכל לזהות את התמונה השגויה ולמחוק אותה). הפרוייקט מקל על המשתמש לעבור ידנית על מקבץ מידע רב בחיפוש לאחר אדם כלשהו, וכן גם מבצע את פעולת החיפוש בצורה מהירה, יעילה ומדויקת יותר. דבר זה חוסך זמן ואנרגיה למשתמש. הפרוייקט מיועד להמון קהלי יעד ביניהם חברות אבטחה שמחפשות אדם חשוב, מישהו שמחפש תמונות שלו עם מכר ועוד המון שימושים... . מה שיפה בפרויקט הזה שהוא פותר בעיה כללית מאוד חשובה כך שיש לו מגוון רחב של שימושים ובעיות שפותר.

חקר מוצרים: מחיפושיי המרובים מצאתי רק פונקציה של גוגל שעושה דבר דומה לשלי. בנוסף, האפליקציה שלי תראה למשתמש את אחוזי ההתאמה בין האדם המבוקש לתמונה בה האדם נמצא, כך שהמשתמש יוכל לבחון תמונות בהם אחוז ההתאמה יחסית נמוך (70%-80%), והאפליקציה שלי (בדומה לגוגל תמונות) תשאל את המשתמש לגבי תמונות בהם אין הכרעה בטוחה לאם התמונה שנמצאה האדם הוא באמת האדם הנכון (75%-90%). בנוסף גוגל תמונות עובר רק על תמונות שנמצאות בגלריית התמונות שלך, ועל כן הוא מוצר לא נוח כי בכל חיפוש תצטרך להוסיף לגלריית התמונות שלך קבצים שאולי לא תרצה שיהיו שם, זה בניגוד לשמירת תיקיות במחשב שם היד של המשתמש יותר חופשית, ובכך האפליקציה שלי יותר נוחה. גוגל תמונות תומך רק במכשירים סלולריים, בניגוד לאפליקציה שלי שמוצעת למשתמשי Windows מה שיותר, לטעמי, נוח ועדיף.

חקר פיתוחי: אני מעריך שני קשיים עיקרים: יישום המכונה שמוצאת פרצופים מתמונות, באופן מדויק והשוואת פרצופים של אנשים שונים. אני מעריך ברגע שקושי אחד ייפתר כך גם השני: כי הרי בשביל למצוא פרצוף אני גם אמצא נתונים על הפרצוף – וכך נפיל שני ציפורים במכה. אני הבנתי שהדרך לממש את הקשיים הנ"ל תהיה בעזרת AI (machine learning), כך שאני אתן למכונה מאגר מידע של תמונות של פרצופים, וכך היא תבין איך לזהות פרצוף, ולהנפיק ממנו את המידע הנחוץ לי בשביל להשוות. האומנם, דבר זה מצריך מבנה נתונים גדול וכוח מעבד גדול ועל כן החלטתי להשתמש במכונה שכבר אומנה – pre-trained.

סביבה : שפת הפיתוח שאני אשתמש בה היא כנראה פייתון בלבד. זאת כי פייתון היא השפה הנוחה ביותר בשבילי, בגלל היותה חופשית יותר משאר השפות. הפרוייקט עתיד כרגע לפעול רק בWindows כי הוא יהיה קובץ exe, בפלטפורמת PC.

תיאור פונקציונלי\גרפי של הפרוייקט המוצע:

**מבנה נתונים-** בפרויקט עתיד להשתלב בסיס נתונים שייכתב באמצעות sql בפייתון. בסיס נתונים זה ישתמש בסוג sqlight3. מבנה הנתונים הפנימי יהיה טבלה, בעלת עמודות ושורות, בה ישמור את הנתונים באופן מאובטח ומסודר. בטבלה ארבעה עמודות: שם משתמש, סיסמא, מיקום ומחובר. בעמודה שם משתמש יהיה שם המשתמש הייחודי של המשתמש, בסיסמא תהיינה סיסמת המשתמש במיקום יהיה מיקום שמירת תמונות המשתמש ובמחובר יהיה כן או לא לפי האם המשתמש מחובר או לא כל שורה תהייה פיסת מידע. הסיבה מאוחרי לשמור את המידע בפיסות מידע הוא כי אין מערכים בsqlight3 ולכן אי אפשר לעשות מערך של התמונות של המשתמש. ובשביל לפתור בעיה זו בדרך היעלה ביותר נאלצתי לפתוח במחשב השרת תיקייה לכל שם משתמש, התיקרא על שמו הייחודית ושם לשמור את התמונות. בשביל לא להגיע למצב שתמונה תישמר עם אותו השם פעמיים אני אוסיף את התאריך והשעה המדויקת של שמירת התמונה על שרת המחשב. בנוסף, אינני רוצה להגיע למצב בו משתמש מחובר בו זמנית למשתמש שלו ולכן יצרתי את עמודת מחובר שתתאפס בכל פעם שהשרת עולה, תתאפס למשתמש שינתק את חיבורו עם השרת ותמנע ממצב כזה להגיע.

|  |  |  |  |
| --- | --- | --- | --- |
| מחובר | מיקום | סיסמא | שם משתמש |
| n | Db\Almog | 1234 | Almog |
| y | Db\Dani1 | Hello123 | Dani1 |

לדוגמא אלמוג נכנס ל"היסטוריה אישית". התוכנה תחפש במיקום Db\Almog את כל התיקיות של האנשים שאלמוג רצה לחפש. תחת כל תיקייה יהיו האנשים שנמצאו להיות מתאים לשם התיקייה והתמונה המקורית של אותו אדם. דוגמא נוספת היא כאשר השרת ייפול כל המשתמשים יהפכו למנותקים וכך לא ינעלו משתמשים. דוגמא נוספת היא שלא תהיינה תיקייה עם אותו שם בשל העובדה שהתיקיות הם לפי שמות המשתמש שהם דבר ייחודי לפי הdb, וגם התמונות יישמרו תחת שם התמונה ותאריך ושעה של תמונה.

**תקשורת-** בפרויקט ישנה תקשורת של שרת-לקוחות. כדי לוודא שהשרת שולח את הpackets ללקוח הנכון אני משתמש בספרייה select.באמצעות ספרייה זו אני יודע איזה לקוח ביקש מה ומי הלקוח העכשווי, שלח את הבקשה האחרונה וכך אני יכול לשלוח לו הודעה בהתאם או לקרוא לפונקציה שתעשה את ההודעה בזמן שאני שולח הודעות ללקוחות אחרים. על מנת ליצור תקשורת הדדית ביניהם אני משתמש במערך שלכל מקום בו יש משמעות. מערך זה הינו ה packet. את ה packet אני שולח באמצעות הספרייה pickle. מבנה המערך יהיה:

|  |  |  |
| --- | --- | --- |
| 2- the sender of the packet (username or Server) | 1- content of the packet (a known contents) | 0- type of the packet |

דוגמא לpackets פוטנציאליות מהלקוח לשרת:

|  |  |  |
| --- | --- | --- |
| "almog123" | "1234" (password) | "login into account" |
| "dani1" | "exit" | "exit" |
| "dani1" | "Hello123" | "create new user" |

דוגמא לpackets שהשרת ישלח ללקוחות:

|  |  |  |
| --- | --- | --- |
| "Server" | "username\_too\_short" | "error" |
| "Server" | "login" | "confirmed" |

על מנת לאבטח את המידע העובר בהודעות האלו אני מבצע "הצפנה סימטרית" (בנוסף לספריה pickle שציינת קודם שגם היא סוג של מצפינה את ההודעות, כדי להעביר סוג של מערך כסוג של בתים). אלגוריתם ההצפנה מבוצע כך שיהיה אפשר להצפין כל הודעה ולפענח כל הודעה כך שאנשים הרואים את ההודעות בין השרת ללקוח לא יצליחו לפענח את ההודעות ואך ורק ללקוח והשרת יהיה את ה"מפתח" (קטע קוד המפענח הודעות מוצפנות) ויוכלו להשתמש בתוכן של הpackets. אני אצפין רק תוכן שהוא קריטי, כלומר שיכול לפגוע בפרטיות המשתמש או לפגוע באפליקציה עצמה.

**תרחישים עיקריים:** בפרויקט ישנם תרחישים עיקרים המחולקים לחלונות שלכל חלון יש מטרה מסויימת, ודרכים להגיע לחלונות ספציפיים אחרים. בפרויקט עתיד להיות ממשק משתמש גרפית באמצעות tkinter. התרחישים העיקרים על פי החלונות בפרויקט הם:

**הרשמה וכניסה-** לאחר הרצת קובץ exe ע"י המשתמש יפתח לו מסך הבית ובו 3 כפתורים: "משתמש חדש", "כניסה" ו"מידע".

לחיצה על לחצן "משתמש חדש" תפתח בפני המשתמש חלון ובו 2 תאי מילוי: משתמש וסיסמא. על שם המשתמש וסיסמא להיות באורך לפחות 4 תווים ועל שם המשתמש לא להיות כבר קיים במערכת. לאחר הרשמה תקינה יחזור המשתמש למסך הבית בו יהיה לו רשום "ההרשמה בוצעה בהצלחה. שלום ותודה שהשתמשת ב'זיהוי פנים'."

לחיצה על כפתור "כניסה" תפתח בפני המשתמש 2 תאי מילוי: משתמש וסיסמא. על שם המשתמש להיות קשור אל הסיסמא ועל שם המשתמש והסיסמא להיות שמורים במערכת. לאחר כניסה תקינה יועבר המשתמש למסך ההוראות.

לחיצה על כפתור "מידע" תפתח בפני המשתמש מידע על האפליקציה: "זיהוי הפנים הינה אפליקציה אשר מאתרת פנים של אדם ספציפי בתוך אוסף של תמונות. האפליקציה שומרת את נתוניך וחיפושך הקודמים. אנא הרשם אם עוד לא עשית זו וכנס למשתמש שלך כדי לגלות עוד...".

בכל חלון בסשן זה יהיה כפתור "חזור" שבאמצעותו המשתמש יוכל לחזור למסך הבית ושמערכת תודיע למשתמש בכל טעות אפשרית.

**הוראות-** לאחר ההתחברות של המשתמש אל החשבון שלו יופיע לו מסך הוראות וכפתור "הבא".

במסך ההוראות יהיה רשום למשתמש איך להשתמש באפליקציה:"1.העלה תמונה של אדם – תן למערכת Path לתמונה של אדם\אנשים מסוים\מים. השתדל שהתמונה תהיה ברורה ככל שאפשר. במקרה של כמה אנשים המערכת תחפש תמונות בהם כל האנשים יחד ולא כל אחד בנפרד. 2.תעלה לאפליקציה תמונות (אתה יכול לתת Path של גלריה מסויימת ואו\בנוסף להוסיף תמונות לפי הPath שלהם). 3. האפליקציה תודיע לך על תמונות אשר לא ידעה להחליט האם האדם בתמונה הוא האדם הרצוי. 4. אם ישנם תמונות אשר הפנים של האנשים בה אינם ברורים סיכויי התמונה למצוא את האדם הרצוי פוחתים. 5. בסוף התהליך התמונות שימצאו לנכון להיות התמונות בהם האדם נמצא ישמרו באופן אוטומטי במשתמש שלך. בנוסף, תהיינה האפשרות להעתיקם לתיקיה אחרת במחשב שלך.

לחיצה על כפתור "הבא" תעביר את המשתמש אל מסך קלט.

בנוסף יהיה כפתור חזור שלחיצה עליו תחזיר את המשתמש אל מסך הבית.

**קלט-** מטרת המסך היא לקבל את המידע הנחוץ לאפליקציה על מנת לפעול- קלט מן המשתמש. בפני המשתמש 3 כפתורים: "אדם\אנשים", "גלריית תמונות" ו"הבא". כמו כן יהיה כפתור "חזור" שיחזיר את המשתמש למסך ה"הוראות".

לחיצה על כפתור "אדם\אנשים" תוביל את המשתמש אל חלון שבו יוצגו כל התמונות שיבדקו ע"י האפליקציה של האנשים שבחר עד כה (כניסה ראשונית אין תמונות) ותא מילוי של Pathים של האדם\אנשים. במקרה זה תהיינה הגבלה של מספר התמונות של עד 3 תמונות (3 אנשים). בנוסף על המערכת לבדוק האם הPath קיים, האם כל הקבצים בו מסוג jpg או png, האם המשתמש העלה תמונה בה אין אדם. המערכת תודיע למשתמש על כל טעות אפשרית. בנוסף יהיה כפתור "חזור" שיחזיר את המשתמש לחלון ה"קלט".

לחיצה על כפתור "גלריית תמונות" תוביל את המשתמש לחלון בו יוצגו כל התמונות שיבדקו ע"י האפליקציה שהעלה עד כה (כניסה ראשונית אין תמונות). פה לא תהיינה הגבלה של כמות או של סוג הקובץ אלא המערכת תודיע מראש שקובץ שאינו מסוג jpg או png, שניתן לעלות עד 2500 תמונות ושתמונה בה אדם לא יזוהה לא תיוחס ע"י המערכת. בנוסף המערכת תבדוק האם הPath לגלריית התמונות קיים. לאחר הכנסת Path תקין יעבור למסך "המתנה לקלט" בו יחכה יהיה רשום "חכה בבקשה שהמערכת תסיים לבדוק את תקינות התמונות...". פה תהיה למשתמש כפתור חזור אשר לחיצה עליו תחזיר את המשתמש לחלון "גלריית התמונות" ותמחק את כל נתונים שנבדקו עד כה (מצב קיצון: המערכת זיהתה 2500 תמונות, במצב זה המערכת תפסיק לעבור על התמונות ותודיע למשתמש על שהפסיקה בשל העלאת יותר מ2500 תמונות). המערכת תראה למשתמש רק את התמונות שיעברו בדיקה בתהליך. בנוסף יהיה כפתור "חזור" שיחזיר את המשתמש לחלון ה"קלט".

לחיצה על כפתור "הבא" תבדוק האם המשתמש הכניס בין 1-3 אנשים לבדיקה. אם לא תודיע לו על כך. אם כן המערכת תעבור למסך ה" המתנה לסיום תהליך הזיהוי".

**המתנה לסיום תהליך הזיהוי-** מסך בו יהיה איזשהו עם מדד גרפי בו יהיה רשום כמה תמונות נבדקו עד כה מתוך כמה תמונות, עם הכיתוב "חכה בבקשה שהמערכת תסיים את התהליך..." ולמטה "לחיצה על כפתור חזור תמחק את כל הבדיקות שבוצעו עד כה ותחזיר אותך לחלון הקלט". המשתמש יהיה תקוע במסך זה עד שהמערכת תסיים את התהליך. בנוסף יהיה למשתמש כפתור "חזור" שלחיצה עליו תחזיר את המשתמש למסך הקלט עם הנתונים עליהם חיפש. כאשר התהליך סוים המשתמש יעבור למסך ה"בדיקה וסיום התהליך".

**בדיקה וסיום התהליך-** במסך זה יהיו 2 כפתורים: "בדיקת תמונות גבוליות" ו"סיום תהליך" ותמונת האדם\אנשים שהמערכת חיפשה (בצד בקטן).

לחיצה על כפתור "תמונות גבוליות" תוביל את המשתמש לייצוג של תמונות שבהם המכונה לא ידעה להחליט בבירור האם האדם בתמונה הינו האדם הרצוי. המכונה מניחה שתמונות אלו הינם האדם הרצוי האומנם עדיף שהמשתמש יתייחס לחלון זה על מנת להקטין את הסיכוי לקליטת תמונה שגויה בהיסטוריית החיפוש האישית של המשתמש. על המשתמש לקבוע האם האדם שבתמונה הגבולית הינו האדם הרצוי. במידה וקבע שלו תמונה זו תוסר מן ההיסטוריה האישית שלו על אותו אדם\אנשים. בנוסף תציג תמונת האדם\אנשים שהמערכת חיפשה (בצד בקטן).

**היסטוריה אישית-** במסך זה יהיה מוצג במעין תיקיות ששמם שם האדם שחיפשת עליו ובכל תיקייה את התמונות שבהם האדם נמצא ותציג תמונת האדם\אנשים שהמערכת חיפשה (בצד בקטן). המשתמש יוכל לעיין בהיסטוריה האישית שלו כרצונו ואף להסיר תמונות .בנוסף תהיינה הודעה למשתמש החיפוש האחרון שלו עבר בהצלחה ונקלט במערכת. במסך זה יהיה כפתור "קלט נוסף" אשר לחיצה עליו תוביל את המשתמש אל מסך ה"קלט". בנוסף יהיה 2 תאי מילוי: "שם התיקייה" ו"Path של גלריה". שבהם יוכל המשתמש להעתיק תיקיות מסוימות מן ה"היסטוריה האישית" שלו לPath במחשב שלו, שהמערכת כמובן תבדוק שPath זה קיים.

\*כל התנתקות תמחק את כל המידע שעוד לא נשמר במערכת ואת כל הקלטים עד כה.

**גבולות הפרוייקט-** לפרויקט ישנם את הגבולות הבאים:

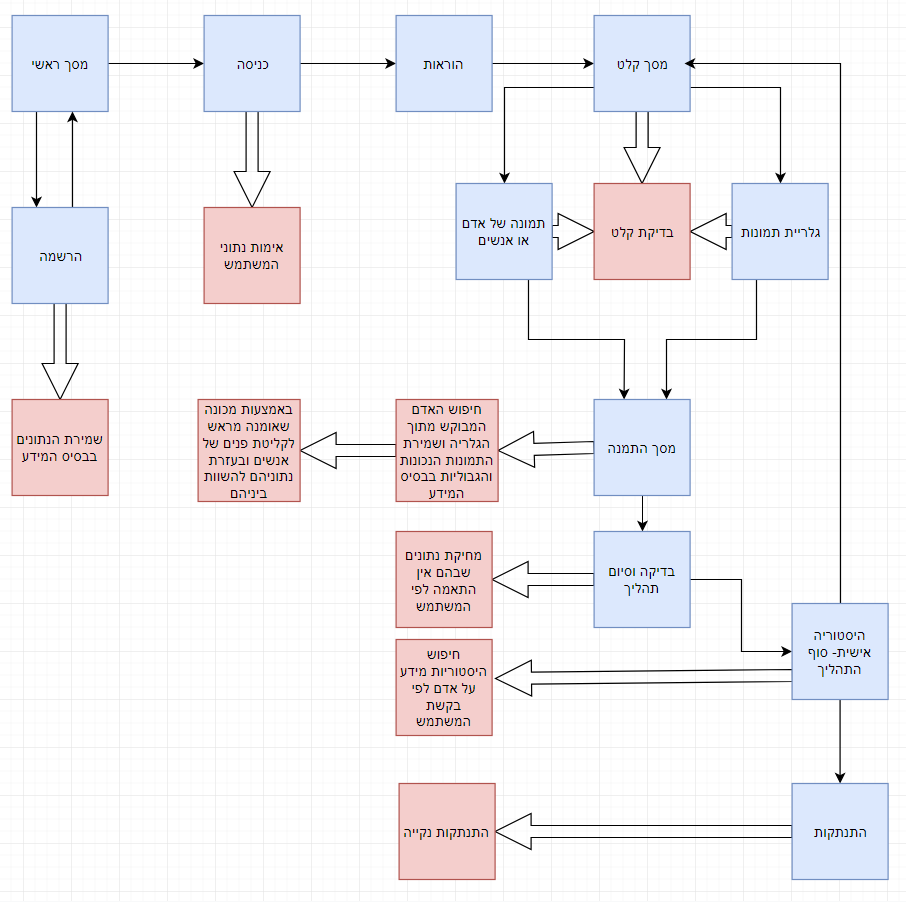
**הגבלת סוג המידע-** הפרויקט יכול למצוא פנים רק בקבצים מסוגים ספציפיים: jpg, png . ועל כן ישנה הגבלה מבחינת המשתמש שצריך לדאוג שהמידע שהוא מעביר לאפליקציה תואם בסוגו.

**הגבלות כוח זיכרון-** מכיוון שתיכנון הdata base שלי צורך הרבה זיכרון, יכול להיווצר מצב של עודף מידע בdata base, כמובן מבחינה תאורטית אם לאפליקציה תהיה רב משתמשים.

**הגבלות כוח מעבד-** האפליקציה תפעל עם threads כך שתוכל לבצע כמה פעולות במקביל, ובכל לטפל בכמה לקוחות במקביל ולא תיתקע בכל בקשה שדורשת כוח מעבד ובכך זמן ריצה רב. ולכן, שוב מבחינה תאורטית, האפליקציה תפעל לאט יחסית משימוש רב משתמשי כי היא מוגבל בכוח מעבד.

**הגבלות מהירות-** יתכן שבגלל הסיבות שציינת לגבי כוח הזיכרון והמעבד יחכו המשתמשים זמן מה בין חלונות, ולכן יש לי גיבוי בחלון ה"המתנה".

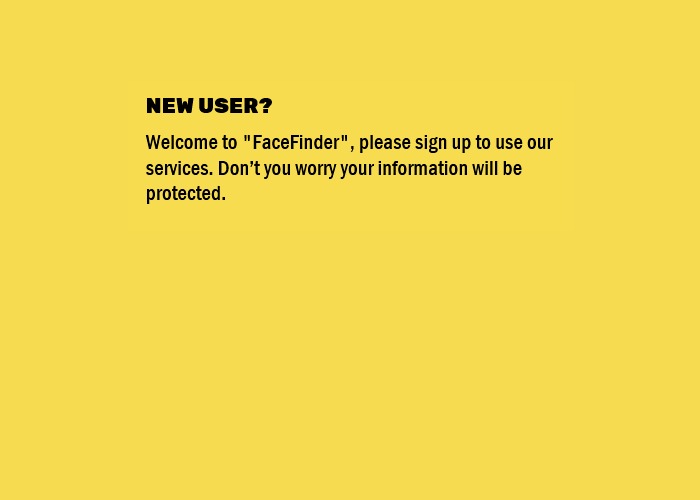
**חלוקות פונקציונליות-**

****

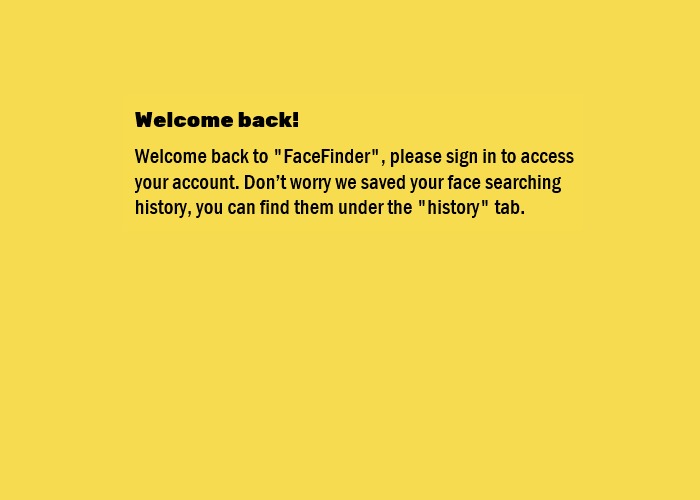
**ממשקים גרפים-** להלן הממשקים הגרפים של התוכנית – תמונות מעוצבות. בנוסף לכל חלון ישנה\ם תמונות error מעוצבות. כרגע אין במסכים אלו כפתורים, כי עדיין לא תיכנתתי אותם.

**מסך ראשי-**

**מסך הרשמה-**

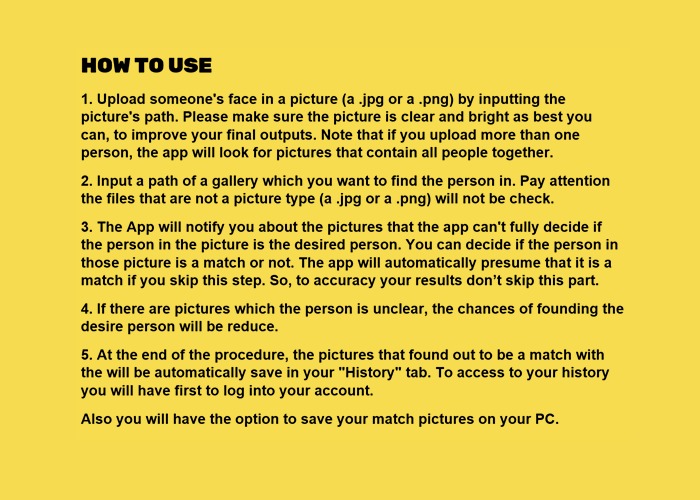
****

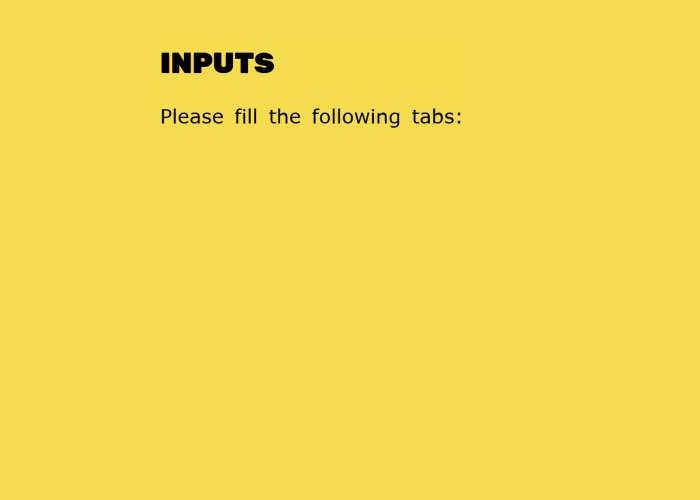
**מסך ראשי לאחר הרשמה-**

****

**מסך כניסה-**

**מסך הוראות-**

**מסך קלט-**

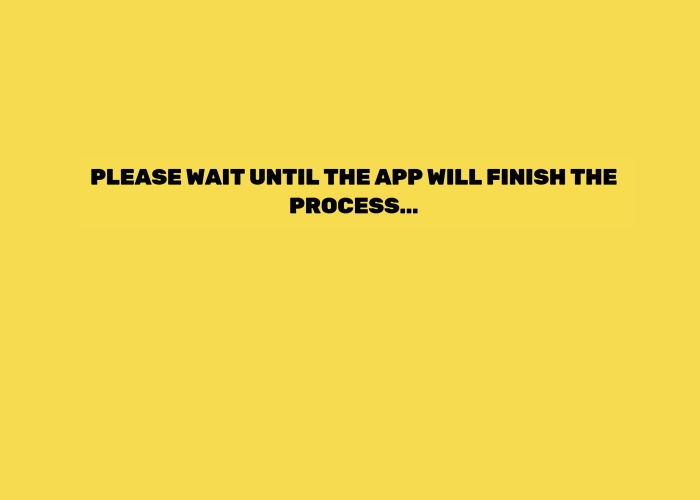
****

**מסך אדם\אנשים-**

****

**מסך גלריית תמונות-**

**מסך המתנה-**

****

**מסך היסטוריה-**

****

**דוגמא למסך error-** אם המשתמש מכניס כקלט שם משתמש שכבר קיים במערכת: