

רשת חברתית – פייקלוק

ברצוננו לבנות רשת חברתית חדשה מבוססת תמונות ומיקום. הרעיון הוא לספק אפשרות ניווט ופילטור של פוסטים המכילים תמונות על ציר זמן, ציר מיקום, ביחס לרשימות חברים ותגיות (ידיניות ואוטומטיות).

ברויקט זה נבנה רשת חברתית דומה לפייסבוק, טוויטר, אינסטגרם, לינקדין וכו'... עם דגש על תמונות ופחות מלל. השאיפה בגלל היקף המידע (כל אדם מצלם כל ג'חנן בכל מסעדה) לפלטר נדבחים מן המידע. המערכת תאפשר בין השאר חיתוך על פי פרמטרים שונים בתבניות גמישות למשל צפייה על גבי מפה כשעליה מופיעים פוסטים שפורסמו ברדיוס גמיש מנקודת ההפעלה של האפליקציה.

המערכת (ככל רשת חברתית) תכלול שירותים בסיסים כגון:

- אימות משתמשים (גוגל, פייסבוק, פוטולוק)
- ניהול רשימות חברים
- התנתקות, הרשמה ואיפוס סיסמא
- העלאת פוסטים – עם תמונה ומיקום (כתנאי קדם להעלאת הפוסטים במערכת הנל)
- ניהול פיד על פי העדפות - סינון פוסטים
- מעקב, חסימה והוספת חברים
- תיוגים בזמן העלאת פוסט וגם בדיעבד,

פונקציונליות ככלית:

- רשת חברתית (חברים/פוסטים וכו').
- הוספת פוסטים - כל פוסט מבוסס מיקום ותמונה כתנאי הכרחי ועם אפשרות לתגיות (מה בתמונה) ותיוגים של משתמשים.
- תמיכה בהוספת תגיות ידיניות ואוטומטיות בכל נקודה במחזור החיים של הפוסט (גם בדיעבד).
- עיבוד וסינון על פי תגיות / זמן / מרחק (רדיוס מהמיקום הנוכחי) / מי פרסם (בודד או רשימה) / מי תווייג (תגיות משתמשים).
- תצוגה על המפה כתמונות ממוזערות.
- שימוש מלא במפה (זום +/-).

מבנה כללי/מסכים מרכזיים/רכיבים בסיסים:

מסך התחברות/הרשמה –

- שדות משתמש וסיסמא.
- כפתור לאיפוס סיסמא
- כפתור להתחברות
- כפתור להרשמה
- כפתור להתחברות עם גוגל
- כפתור להתחברות עם פייסבוק


Login


Username

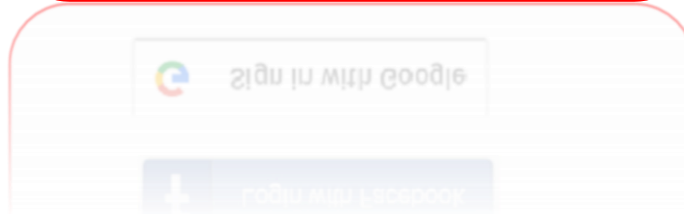
Password

[forgot password?](#)

Not yet a member? [Sign up](#)

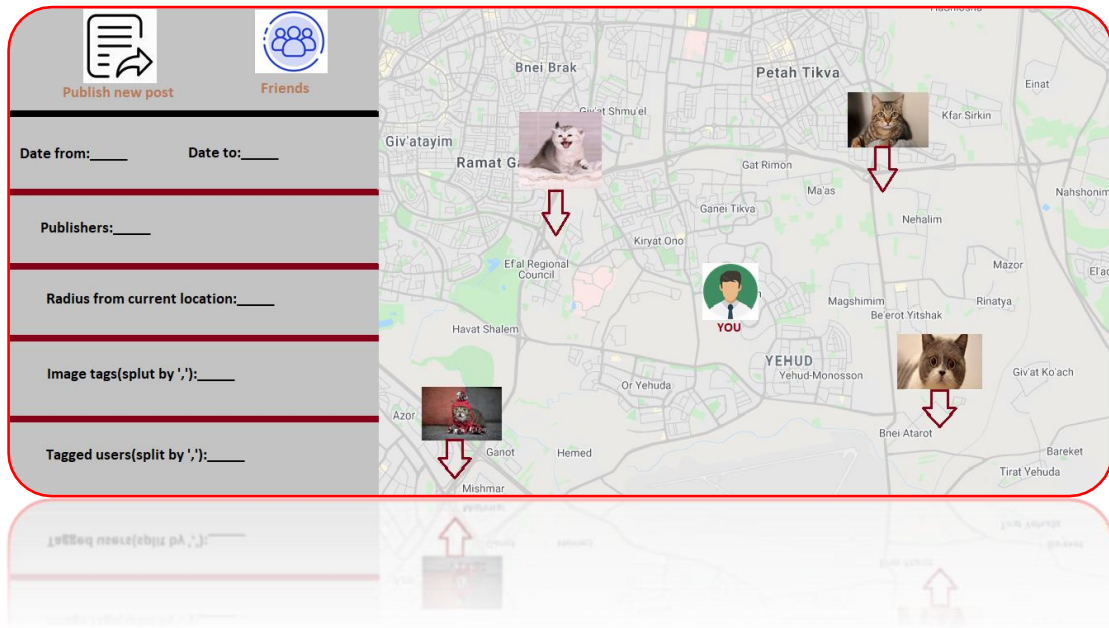
 Login with Facebook

 Sign in with Google



מסך ראשי –

- צד ימין יתפוס את רוב המסך ויצג מפה שעליה מופיעים הפוסטים כתמונות ממוזערות.
- המפה תציג את המיקום של המשתמש ברדיוס של X קילומטר ותתמוך בהגדלה והקטנה (זום אין/זום אאוט).
- בזמן הרחבת או הקטנת שטח המפה תעודכן התצוגה בפוסטים חדשים במידה וישנם כאלו מתאימים לרזולוציה החדשה.
- לחיצה על פוסט תציג את הפוסט במלואו כתפריט צף (תגיות, תיוגים ומפרסם).
- הצד השמאלי של המסך יכיל בחלק העליון שלו סרגל כלים עם לוגיקת גוי לניהול חברים ולפרסום פוסט חדש.
- מתחת לסרגל כלים יהיה תפריט של סינונים דינמיים ככל האפשר. ביניהם רדיוס, טווח תאריכים, חבר, בחירת תגיות ובחירת "מעגל חברים" (כפי שיוגדר בהמשך) לתצוגה.

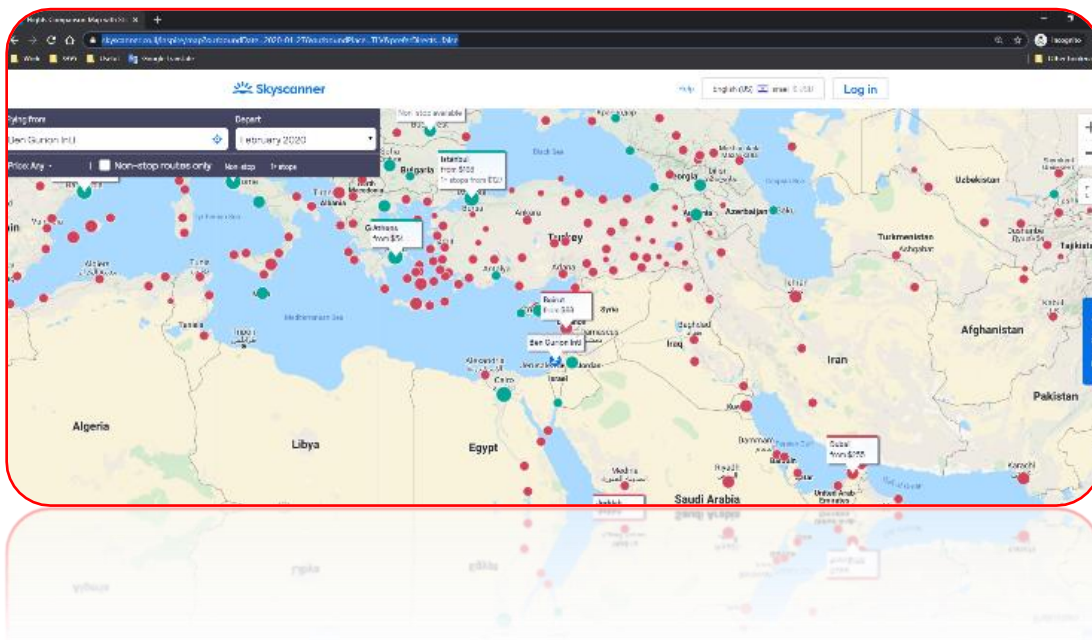


ניהול חברים –

- מסך זה יחולק לשני אזוריים אזור החברים ואזור תתי הקבוצות
- רשימה של קשרים עם חברים עם אפשרויות
 - מחיקה של חבר.
 - יצירת תת קבוצות חברים – "מעגל חברים" (לצורך פילטור)
 - כל תת קבוצה תכיל לפחות חבר אחד
 - לכל תת קבוצה יהיה שם ייחודי.
 - עריכת תת קבוצות
 - מחיקה של תת קבוצה
 - הסרת חברים \ הוספת חברים לתת הקבוצה
- הוספת חברים לקבוצה תתבצע ב drag and drop

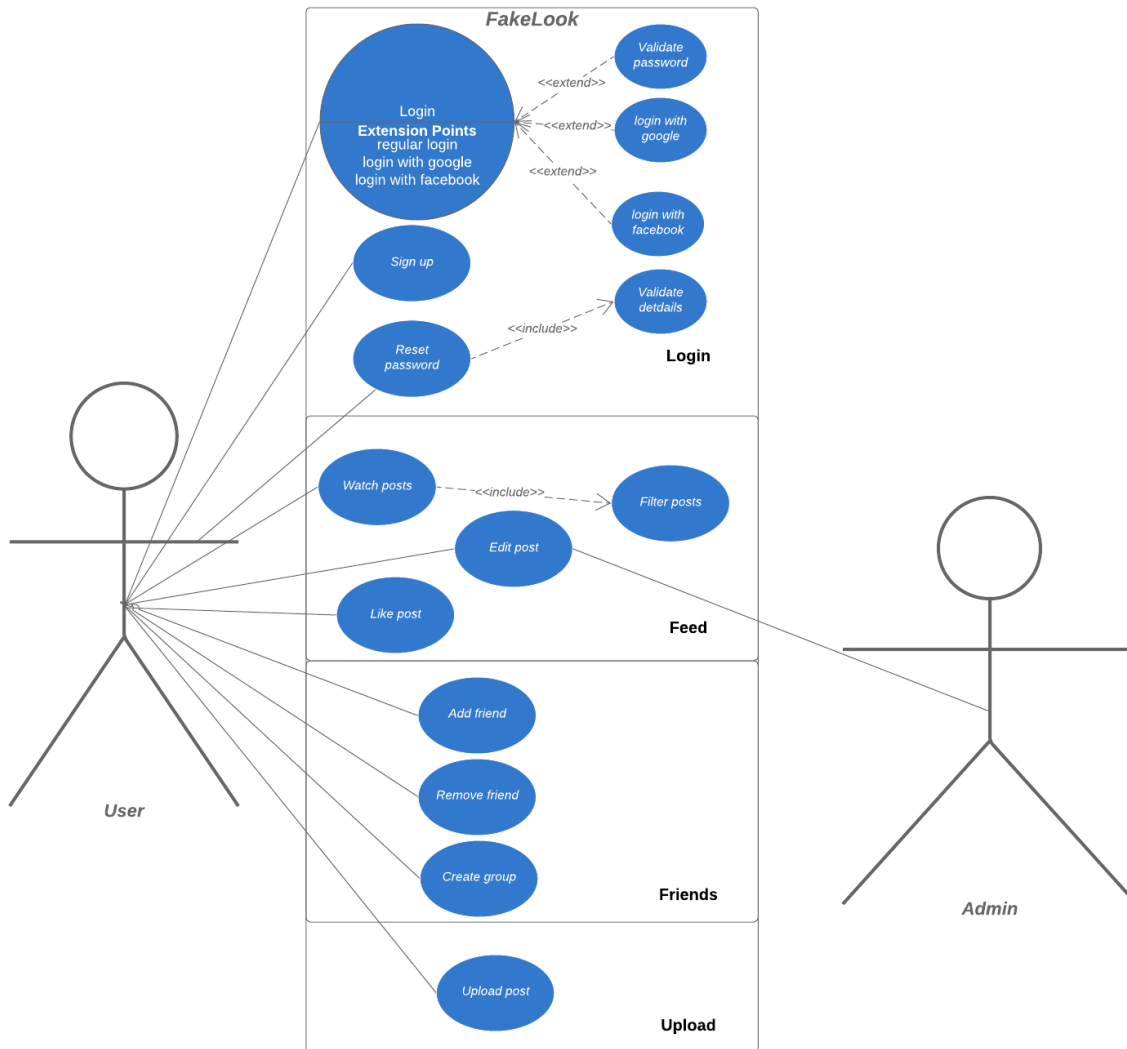
תצוגת מפה -

- המפה תוצג בצורה דינאמית לחלוטין.
- גלילת עכבר ואירוע מקלדת מקביל לזום פנימה והחוצה
- עדכון מפה דינאמי בהתאם
- עדכון שכבות על המפה בהתאם לשינוי המפה בזמן אמת
- בחירת רזולוציית תמונות לתצוגה
 - נאמר שעל המפה יהיו בין 0 – 100 תמונות פוסטים (במקרה שיש)
 - היה ויש פוסטים נוספים יבחרו אילו עם המתאם הגבוהה ביותר לחיפוש למשל על פי מספר הלייקים שכל פוסט קיבל. פוסטים אחרים יוצגו כנקודות.
- תגובתיות המפה ועדכונה מבחינת zoom in \ zoom out צריכה להיות מקבילה ל
- <https://www.skyscanner.co.il/inspire/map>



יצירת פוסט חדש – (יופיע מכל מסך פנימי בלחיצת כפתור כ popup window)

- העלאת תמונה מתוך קובץ בשלב זה ללא חיבור למצלמה.
- תיוג משתמשים
- הוספת תגיות על התמונה
- המיקום של התמונה ישמר אוטומטית לפי מיקום העלאה.
- תאריך ומפרסם הפוסט ישמרו אוטומטית



שלב א : מערכת בסיסית (שימוש באנגולאר ו Node js , SQL)
שלב ב : מעבר לארכיטקטורה (Microservices , Docker , K8S , ElasticSearch ובדיקות ב swagger)
שלב ג: שילוב כלים , GIS ובסיסי נתונים נוספים

תהליך הפיתוח

- הפיתוח יהיה בקבוצה של בודדים או זוגות על פי החלטת מנחה הפרוייקט.
- הקוד יאוחסן ב git repository
- הצוות יתכנן את הפרוייקט לפני המימוש
 - ארכיטקטורה ו Physical Design יתועדו במסמך
 - Interfaces
 - Models
 - High level Architecture (מי פונה למי מתי וכמה)
 - Milestones
- כל פיצ'ר יפותח מתחילתו ועד סופו במסגרת העבודה יש להקפיד ללא יוצא מן הכלל על תיעוד כל שלב:
 - עדכון ה Physical Design
 - המימוש ב Database
 - מימוש הלוגיקה
 - מימוש הבדיקות
 - מימוש ה Service
 - מימוש ה UI
 - עדכונים הדדים
- כל פיצ'ר שהפיתוח שלו הסתיים יעלה לסביבת ה QA ויוכל להיבדק כל Milestone יעלה ל Production.
- תשומת לב מיוחדת תהיה על הפרדה מלאה בין ה services, על מנת לעבוד עם עקרונות של microservices. הנתונים יעברו בין ה services בעזרת REST/Grpc וכו' בין הלקוח לשרת תהיה נקודת גישה אחת ואז מהסרבר הראשי תהיה גישה לכל ה services האחרים כלומר ממנו תהיה תקשורת ל service שאחראי ל DB

שלב א : מערכת בסיסית (שימוש באנגולאר ו Node js , SQL וממשקים)

- המערכת תכלול שירותים מגוונים וביניהם:
 - Authentication Service
 - התחברות בעזרת שם משתמש וסיסמא.
 - התחברות בעזרת Facebook.
 - מנגנון איפוס סיסמא.
 - יציאה וניתוק חיבור .
 - תיקוף חיבור –
 - לאחר התחברות השירות יחזיר Token שיאפשר לפנות לכל השירותים בטווח 15 דקות מהפעולה האחרונה שנעשתה.
 - רכיב זה יעבוד עם AWS DynamoDB
 - Identity Service
 - ישמש לשמירת לשמור פרטי משתמש :
 - שם מלא
 - גיל
 - כתובת
 - מקום עבודה
 - רכיב זה יעבוד עם AWS DynamoDB
 - בכל פנייה יתבצע מעבר דרך Token ותהליך ולידציה סביר.
 - אפשרות ליצירת פוסט מבוסס מיקום ותמונה .
 - אפשרות ליצירת פוסטים שבתוכן יתכנו
 - טקסט
 - תמונה - חובה
 - תגיות
 - מיקום - חובה
 - אפשרות לביצוע לייק לפוסטים או תגובות
 - אפשרות להוספת תגובה לפוסט שיכולה להכיל
 - טקסט
 - אזכור של משתמש אחר
 - תגיות חדשות
 - הצגת הפוסטים בשני מצבים , בפיד רציף מבוסס זמן או על גבי מפה
 - הפיד הרציף
 - על הפיד יוצגו כלל הפוסטים של כלל המשתמשים במערכת בשרשרת על פי סדר הפירסום
 - פוסט חדש שעולה לאויר יעדכן את הפיד בזמן אמת וישורשר לראשו
 - כל פוסט יוצג בתקציר (תמונה ממוזערת ומלל אם קיים) ובלחיצת כפתור יורחב לשאר הפרטים ותמונה בגודל קבוע / יחסי לתצוגה ואת פרטי הפוסט.
 - הפיד הגרפי
 - על פני מפה יוצגו כלל הפוסטים שפורסמו
 - המפה תהיה בגודל קבוע מבחינת אורך ורוחב
 - תכולת (גבולות) המפה יקבעו על פי zoom in / zoom out (באירוע עכבר ו מקלדת) ובכל מקרה תמיד תהיה ממורכזת פחות או יותר על פי המיקום הנוכחי של המשתמש .
 - אם בגבולות הגזרה נאמר שיש X פוסטים אזי -
 - אם $30 > X$ פוסטים יוצגו במיקום שלהם על המפה כתמונות ממוזערות
 - אחרת אם $100 > X$ פוסטים יוצגו כנקודות / תמונת נעץ
 - אחרת אם $100 < X$ יוצגו 100 הפוסטים שיש להם מספיר like ים הגדול ביותר.
 - בלחיצה על פוסט יפתח תפריט צף שיציג את התמונה בגודל קבוע / יחסי לתצוגה ואת פרטי הפוסט.

שלב ב : השלמת הרשת החברתית (Microservices)

המערכת תכלול שירותים מגוונים ובינם:

- Social Service
 - יצירת מעגלים (קבוצות שייכות)
 - אפשרות וחסומת משתמש -
 - היה ומשתמש X נחסם על ידי משתמש Y אזי משתמש X לא יוכל לראות את התכנים של משתנה Y ל X לא שמורה הזכות לפרטיות, מידע שיעלה יהיה חשוף לכלל המשתמשים.
 - היה ומשתמש X נחסם על ידי משתמש Y הוא יגרע כמובן מכל המעגלים אליהם היה שייך בפרופיל של משתנה Y, אם הוא שייך למעגל בו הוא המשתתף היחיד המעגל ימחק.
 - צפייה בפרופיל משתמש מרשימת החברים.
 - יצירת מעגלים (קבוצות שייכות)
 - אפשרות ליצירת תתי קבוצות (חבורות/מעגלים) שישימשו כאופציה לממשק תצוגה (פילטור) על המסך, אפיון זה יהיה ידני ויאפשר הצגת תמונות על פי החתכים הללו
 - אפשרות לקביעת הרשאות צפייה לפוסט
 - כולם
 - מי עוקב אחרי
 - בכל פנייה יתבצע מעבר דרך Token ותהליך ולידציה סביר.
 -
 - Notification Service
 - אפשרות לצפייה בהודעות על פעילות הקשורה למשתמש
 - אינדיקציה לזה שממישהו עשה לייק למשתמש , לפוסטים או לתגובות שלו.
 - קבלת ההודעות בזמן אמת
 - בכל פנייה יתבצע מעבר דרך Token ותהליך ולידציה סביר.
 - רכיב זה ישתמש ב XMPP
 - שילוב יכולות GIS – יפורט בהמשך
 - שילוב יכולות Elasticsearch – יפורט בהמשך
-

שלב ג : הפיכת המוצר לסקלבילי שימוש ב REDIS ויזואליזציה בקרה ואנליזה (Kibana , Metrics ...)

– יפורט בהמשך