**פרויקט גמר – סימסטר א':**

**שם הפרויקט: PAWsitive Life**

**שמות אנשי הצוות:**

שם: אור אביכזר אלמליח שם: נועה דנון

ת.ז: 318659455 ת.ז: 207229303

**מטלה 3 – Detailed Design:**

עיצוב מפורט – Detailed Design:

A diagram of a application

Description automatically generated**דיאגרמת רצף (Sequence Diagram) – תיאור הזרימה:**

**1. הוספת פרופיל כלב:**

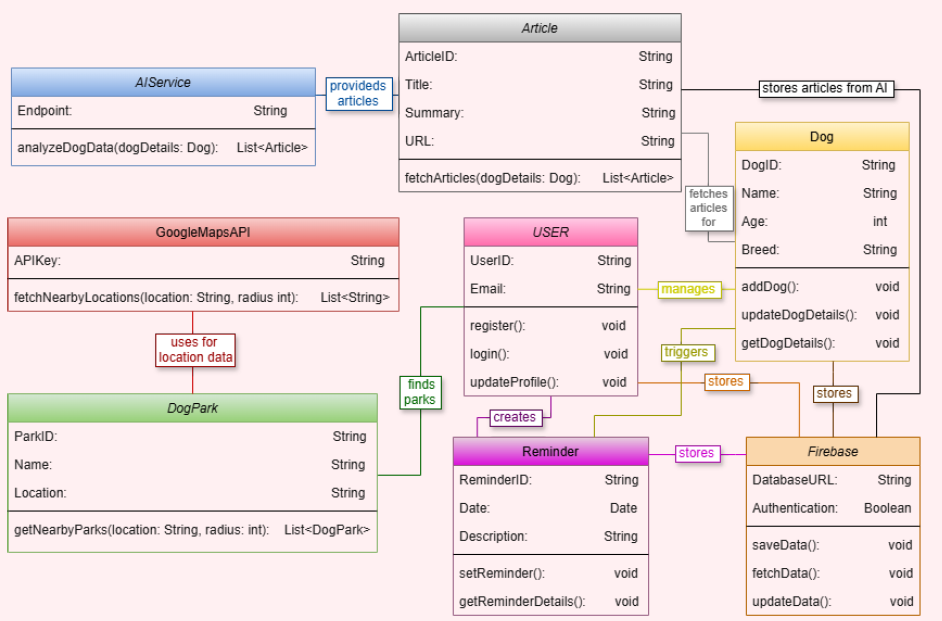
* משתמש מוסיף פרטי כלב (שם, גיל, גזע וכו') דרך ממשק האפליקציה.
* האפליקציה שולחת את המידע ל-Firebase, שם נשמרים הנתונים בצורה מאובטחת ובזמן אמת.

**2. בקשת מאמרים ומידע מותאמים אישית:**

* משתמש מבקש מאמרים הקשורים לטיפול בכלבו.
* האפליקציה מעבירה את הבקשה ל-API Gateway, שתפקידו לתקשר עם שירות ה-AI החיצוני.
* AI Service מנתח את פרטי הכלב (גיל, גזע וכו׳) ומחזיר מאמרים מותאמים אישית.
* המאמרים נשלחים דרך ה- API Gateway חזרה לאפליקציה ומוצגים למשתמש.

**3. מציאת גינות כלבים קרובות:**

* משתמש מבקש למצוא גינות כלבים קרובות.
* האפליקציה שולחת את מיקום המשתמש ל- Google Maps API יחד עם רדיוס חיפוש.
* Google Maps API מחזיר נתונים על גינות קרובות, כולל כתובות ומידע רלוונטי.

**דיאגרמת מחלקות (Class Diagram):**

**1. מחלקות עיקריות:**

* :User מחלקה לייצוג המשתמשים באפליקציה.
* Dog: מחלקה לניהול פרטי כלבים.
* :Reminder מחלקה לניהול תזכורות.
* :Article מחלקה לניהול והצגת מאמרים מותאמים אישית.
* :DogPark מחלקה אחראית על מציאת גינות כלבים קרובות.

**2. קשרים בין מחלקות:**

* User מנהל פרופילים של Dog ויוצר Reminder.
* Dog קשור למאמרים (Article) ותזכורות (Reminder).
* DogPark מתקשר עם GoogleMapsAPI כדי לאתר גינות קרובות.

**3. שירותים חיצוניים:**

* :Firebase לאחסון נתונים.
* :AIService משמש לאחזור מאמרים מותאמים אישית.
* :GoogleMapsAPI משמש למציאת גינות קרובות.

**רכיבים מרכזיים ותפקידיהם:**

* משתמש (User): פועל דרך האפליקציה לביצוע פעולות כמו הוספת פרופיל, בקשת מאמרים, ומציאת גינות קרובות.
* אפליקציית אנדרואיד (Android App): מנהלת את האינטראקציה עם המשתמש ומתווכת בין המשתמש למערכת.
* Firebase: משמש כמסד נתונים ומאחסן נתוני פרופיל, תזכורות, והגדרות משתמשים.
* API Gateway: מתווך בין האפליקציה לשירותי AI חיצוניים.
* AI Service: מבצע ניתוח של פרטי הכלב ומחזיר מאמרים מותאמים אישית.
* Google Maps API: מספק נתוני מיקום רלוונטיים, כמו גינות כלבים קרובות או וטרינרים.

**עיצוב המערכת הכוללת:**

המערכת מבוססת על ארכיטקטורת **MVC (Model-View-Controller)**, שמארגנת את המערכת לשכבות נפרדות:

* **Model (מודל):** מנהל את לוגיקת הנתונים (כגון פרופילי כלבים ותזכורות).
* **View (תצוגה):** ממשק המשתמש באפליקציה.
* **Controller (בקר):** אחראי על ניהול התקשורת בין המודל לתצוגה, כולל קריאות ל-API ושימוש ב-Firebase.

**אלמנטים של עיצוב מפורט:**

**א. זרימת נתונים:**

נתונים מועברים מהמשתמש לאפליקציה ומשם:

* ל-Firebase עבור אחסון נתונים וקריאה שלהם בזמן אמת.
* ל- API Gateway לשילוב שירותי AI חיצוניים.
* ל- Google Maps API לקבלת נתוני מיקום.

**ב. רכיבים עיקריים:**

1. אפליקציית אנדרואיד:

* הממשק העיקרי בין המשתמש למערכת.
* מנהלת את הקריאות ל-Firebase, ל-API Gateway, ולשירותי מפות.

1. Firebase:

* מספק אחסון נתונים ותמיכה בזמן אמת.
* מאחסן פרטי משתמשים, פרופילי כלבים, ותזכורות.

1. AI Service:

* מנתח את פרטי הכלבים (גזע, גיל וכו') ומחזיר מאמרים מותאמים אישית.
* תומך בממשק REST עבור תקשורת עם ה-API Gateway.

1. Google Maps API:

* מספק מידע גיאוגרפי כמו מיקום גינות קרובות.

A screenshot of a computer

Description automatically generated

1. **חבילות (Packages):**

* User Interface: מייצגת את ממשק המשתמש.
* Backend Services: מייצגת את השירותים האחוריים, כולל Firebase ו-API Gateway.
* :External Services מייצגת שירותים חיצוניים, כמו Google Maps API ו-AI Service.

1. **רכיבים (Components):**

* :User המשתמש שמבצע פעולות באפליקציה.
* :Android App ממשק המשתמש המרכזי של המערכת.
* :Firebase לאחסון נתונים ולניהול תזכורות.
* :API Gateway מתווך בין האפליקציה לשירותי AI.
* :AI Service מספק מאמרים מותאמים אישית.
* Google Maps API: מספק מידע על מיקומים קרובים.

1. **קשרים (Arrows):**

* החצים מראים את זרימת המידע בין הרכיבים, כולל שליחת בקשות וקבלת תשובות.

**יתרונות העיצוב:**

**1. סקלאביליות:**

* **Firebase** מספק תמיכה בסיסית בניהול נתונים ובקריאות בו-זמניות.
* **API Gateway** מאפשר שילוב של שירותים חיצוניים נוספים בעתיד. (?)

**2. מודולריות:**

* כל רכיב מופרד בצורה ברורה, מה שמאפשר תחזוקה ושדרוגים קלים.

**3. התמקדות במשתמש:**

* מתן עדכונים בזמן אמת, מאמרים מותאמים אישית, וגישה לנתוני מיקום משפרים את חוויית המשתמש.

**4. גמישות לעתיד:**

* ניתן להוסיף פיצ'רים חדשים (כגון טיפים לאילוף כלבים) מבלי להשפיע על המערכת הקיימת.

**דוח יישום וכלי פיתוח:**

1. **מערכת לניהול גרסאות (Version Control System):**

* כלי: Git עם GitHub.
* שימוש:
  + ניהול גרסאות קוד, יצירת ענפים לפיתוח תכונות ושמירת שינויים.
  + GitHub ישמש כמרכז גיבוי ושיתוף פעולה.

1. **סביבת פיתוח (IDE):**

* כלי מרכזיAndroid Studio :.
* תוספות חשובות:
  + Firebase Assistant להגדרת שירותים.
  + Git Integration לניהול גרסאות.
  + Lint לשיפור איכות הקוד.

1. **כלי בנייה (Build Tools):**

* כלי מרכזי:Gradle .
* שימוש:
  + ניהול תלות (dependencies)ובניית קובץ APK (קובץ האפליקציה). (?) (APK באמצעות אנדרואיד סטודיו ללא אוטומציה מתקדמת)
  + הבנייה תבוצע ידנית מתוך Android Studio.

1. **כלי דיבוג (Debugging Tools):**

* כלים:
  + Logcat לניטור הודעות ושגיאות בזמן אמת.
  + Firebase Crashlytics אופציונלי, לאיתור בעיות וקריסות אם הזמן יתאפשר. (?)

1. **שירותי צד-שרת (Backend Services):**

* Firebase:
  + אחסון וניהול נתוני משתמשים בזמן אמת.
* Google Maps API:
  + מתן מידע על גינות כלבים קרובות.

**דרישות:**

* משתמשים יכולים להירשם וליצור חשבון.
* משתמשים יכולים להוסיף פרופילים של כלבים עם פרטים כמו שם, גיל, גזע ועוד.
* משתמשים מקבלים מאמרים וטיפים מותאמים אישית לטיפול בכלבם.
* משתמשים יכולים לקבל תזכורות לחיסונים וטיפולים רפואיים.
* משתמשים יכולים למצוא גינות כלבים קרובות באמצעות מיקום.

**טכנולוגיות:**

* **Frontend:** אפליקציית אנדרואיד ב-Kotlin.
* **Backend:** Firebase עבור אחסון נתונים בזמן אמת ואימות משתמשים.
* **שירותי צד שלישי:** 
  + Google Maps API לאיתור גינות כלבים קרובות.
  + OpenAI API לקבלת מאמרים מותאמים אישית.

**רכיבים:**

* **אימות משתמשים:**

שימוש ב-Firebase Authentication לאימות ואבטחת חשבונות.

* **ניהול פרופיל כלב:**

ממשק להוספה, עריכה ומחיקה של פרופילי כלבים.

* **מאמרים מותאמים אישית:**

קריאה ל-API של OpenAI לקבלת תוכן מותאם לפי פרטי הכלב.

* **תזכורות:**

מערכת תזכורות לחיסונים וטיפולים, עם התראות פוש (Push Notifications).

* **מציאת גינות קרובות:**

שימוש ב-Google Maps API להצגת גינות כלבים על מפה.

* **ממשק מנהל (עתידי):**

אפשרות לעדכון וניהול תוכן המאמרים והתזכורות.

**ציר זמן:**

* **שבוע 1-2: הקמת פרויקט והגדרת תשתיות:** 
  + התקנת Android Studio והגדרת סביבת פיתוח.
  + חיבור ל-Firebase והגדרת Authentication.
  + יצירת שלד בסיסי של האפליקציה.
* **שבוע 3-4: פיתוח Backend ו-API:**
  + הגדרת מבנה הנתונים ב-Firebase.
  + פיתוח פונקציות לאחסון ושליפת נתונים (פרופילי כלבים, תזכורות).
  + אינטגרציה עם OpenAI API לקבלת מאמרים.
* **שבוע 5-6: מימוש רכיבי Frontend:**
  + עיצוב ופיתוח ממשקי משתמש (UI) לפרופיל משתמש וכלב.
  + פיתוח מערכת התזכורות וההתראות.
  + אינטגרציה עם Google Maps API להצגת גינות קרובות.
* **שבוע 7: בדיקות ודיבוג:** 
  + ביצוע בדיקות יחידה ובדיקות שילוב.
  + תיקון באגים ושיפור חוויית המשתמש.
  + בדיקת אינטגרציה מלאה בין כל הרכיבים.
* **שבוע 8: הפצה ותיעוד:** 
  + הכנת האפליקציה להפצה (יצירת APK).
  + כתיבת מדריך משתמש ומסמכי תיעוד.
  + הצגת הפרויקט וביצוע הדגמה.

**תיעוד:**

* **README:** הוראות התקנה והפעלה של האפליקציה.
* **תיעוד API:** תיאור הקריאות ל-OpenAI API ו-Google Maps API.
* **מסמכי עיצוב:** דיאגרמות UML, תרשימי זרימה ומבנה בסיס הנתונים.
* **מדריך משתמש:** הסברים על שימוש באפליקציה ותכונותיה.

**סיכום:**

הפרויקט PAWsitive Life מתמקד במתן פתרון כולל לבעלי כלבים, המשלב ניהול מידע, תוכן מותאם אישית, ותכונות מבוססות מיקום. באמצעות שימוש בטכנולוגיות מודרניות ותכנון מובנה, האפליקציה תשפר את חוויית המשתמש ותספק ערך מוסף לקהל היעד.