### תיאור קצר של הפיצ'רים שבחרנו לממש בתרגיל הקודם

**פיצ'ר מס' 1 – "FaceRide!":**

פיצ'ר המאפשר למשתמש למצוא טרמפ מכל מקום לכל מקום בעזרת הרשת החברתית.

כדי להשתמש בפי'צר הנ"ל, על המתמש לבצע:

1. הזנה למערכת את נקודת האיסוף, או לחילופין שימוש במיקמו הנוכחי של המשתמש בעזרת שירותי מיקום.

2. הזנה למערכת את היעד, אותו יוכל להזין באופן ידני או "למשוך" מתוך רשימת האירועים אליו הוא רשום. לאחר מכן יתאפשר למשתמש לצפות ביעד בו בחר במפה בחלון נפרד.

3. בחירת רדיוס חיפוש רצוי ומגדר.

המערכת מאתרת מרשימת חבריו של המשתמש את כל אלו אשר נמצאים ברדיוס החיפוש בו בחר ומציגה אותם כרשימה המכילה מידע בסיסי אודותם. בעת בחירת המשתמש (לחיצה כפולה על החבר הרצוי), נפתח מסך ובו הודעה Auto Generated המוכנה לפרסום על הקיר או לשליחה דרך ה-Messenger, המציגה את בקשת המשתמש מאותו חבר לטרמפ אל היעד אליו הוא רוצה להגיע.

**מחלקות המעורבות בפיצ'ר**:

* AppForm
* FaceRideManager
* LocationServices
* AppFacade
* MapForm

**פיצ'ר מס' 2 – "Contacts":**

פיצ'ר זה נועד ליצור רשימת אנשי קשר מנתוני חברי הפייסבוק של המשמש ויצוא הרשימה לסמרטפון של המשתמש.

דרך הפעולה:

1. על המשתמש לעבר ללשונית הcontacts.

.2 לאחר מכן על המשתמש ללחוץ על כפתור הshow.

3 המערכת תיצור רשימת אנשי קשר ותזין אוטומטית את הנתונים לטבלה וויזואלית.

4. אופציונלי: המשתמש יכול לערוך בטבלה את נתוני החברים מהפייסבוק שהוזנו לפי טעמו.

5. אופציונלי: המשתמש יכול לערוך עמודות נוספות רלוונטיות לרשימת אנשי קשר שאינן חלק מנתוני הפייסבוק אך הן כן רלוונטית בשביל אנשי הקשר, חלק מעמודות אלו מותאמות בשמותיהן לתצוגה מיוחדת במכשרי הסמרטפון.

6. על המשתמש ללחוץ על כפתור הsave כדי לשמור את הנתונים כקובץ.csv

.7 בלשונית תפתח דפדפן פנימי אשר נמצא בדף הסבר מיוחד על איך לייצא את רשימת אנשי הקשר באופן מיידי לחשבון Gmail שיאפשר שימוש פרקטי ברשימת אנשי הקשר שנוצרה בסלולרי האישי של המשתמש.

**מחלקות המעורבות בפיצ'ר**:

* AppForm
* AppFacade
* GoogleContacts (המכיל תתי מחלקותEmail, Organization, Websites, Address, Phone )

### תבנית מס' 1 –Strategy

* **סיבת הבחירה / שימוש בתבנית:**

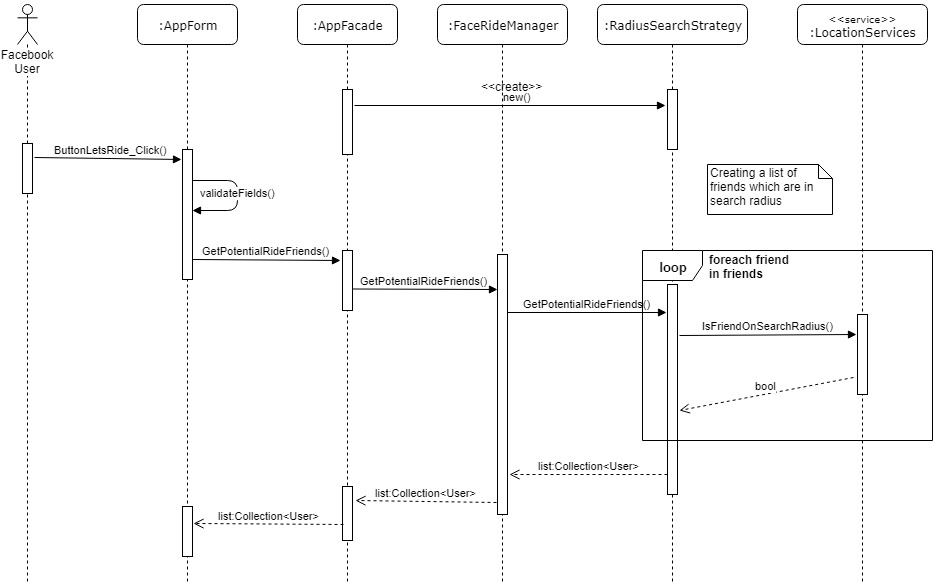
באפליקציה שלנו קיים פיצ'ר המאפשר איתור של חברים פוטנציאלים לנסיעה משותפת.

המערכת כרגע מקבלת רדיוס חיפוש רצוי מהמשתמש ועל פיו מאתרת את החברים האפשריים לנסיעה. אם בעתיד, נרצה להרחיב את הפיצ'ר ולאפשר לבנות את רשימת החברים הפוטנציאלים לנסיעה על פי חיפוש אחר שאינו קשור במרחק ו/או בשימוש ברדיוס חיפוש, נוכל לעשות זאת ללא שיכפול קוד או תחזוקה מיותרת של קוד קיים ע"י החלפת הרכיב SearchStrategy, שהוא Property אסטרטגי של המחלקה FaceRideManager.

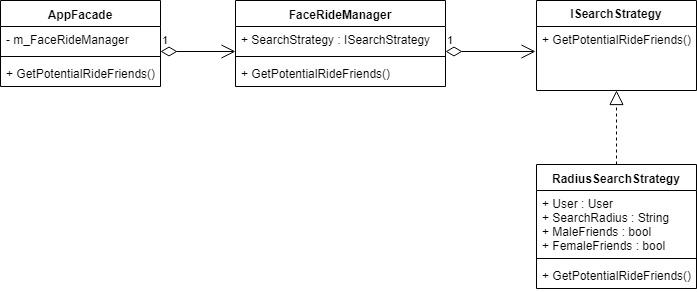
* **אופן המימוש:**

יצרנו Interface שנקרא ISearchStrategy הממוקם בתיקיית Interfaces תחת הפרוייקט Model. כאמור, המחלקה FaceRideManager מחזיקה בתור Property אובייקט אסטרטגיה מהסוג ISearchStrategy וכאשר נדרשת מה-Client לקבל את רשימת החברים האפשריים לנסיעה בטרמפ, היא מפעילה את אסטרטגית החיפוש בה בחר ה-Client מבעוד מועד, באופן המאפשר שינוי אסטרטגיה בזמן ריצה.

* **:Sequence Diagram**



* **Class Diagram:**



### תבנית מס' 2 – [שם התבנית]

* סיבת הבחירה / שימוש בתבנית:

[תיאור הסיבה / הצורך בשימוש בתבנית במערכת שלכם]

* אופן המימוש:

[תיאור המימוש והיכן ניתן למצוא אותו בקוד]

* Sequence Diagram

[sequence diagram שמציגה את התהליך שקשור לתבנית והאינטראקציה בין המחלקות]

* Class Diagram

[class diagram שמתארת את המחלקות שלכם שמעורבות בתבנית (תיאור מלא שכולל Properties ו- Methods) והיחסים ביניהם (תאור מלא עבור כל יחס כפי שלמדנו בכיתה). עבור כל מחלקה שלכם, כיתבו מי המקבילה שלה (אם יש כזו) בתבנית (Pattern) שבחרתם

### תבנית מס' 3 – [שם התבנית]

* סיבת הבחירה / שימוש בתבנית:

[תיאור הסיבה / הצורך בשימוש בתבנית במערכת שלכם]

* אופן המימוש:

[תיאור המימוש והיכן ניתן למצוא אותו בקוד]

* Sequence Diagram

[sequence diagram שמציגה את התהליך שקשור לתבנית והאינטראקציה בין המחלקות]

* Class Diagram

[class diagram שמתארת את המחלקות שלכם שמעורבות בתבנית (תיאור מלא שכולל Properties ו- Methods) והיחסים ביניהם (תאור מלא עבור כל יחס כפי שלמדנו בכיתה). עבור כל מחלקה שלכם, כיתבו מי המקבילה שלה (אם יש כזו) בתבנית (Pattern) שבחרתם