

## מסמך ניתוח - ShareIT

שמות חברי הקבוצה:

תומר שור - 325511541

אוראל ניסן - 322861527

עוז עטר - 214396152

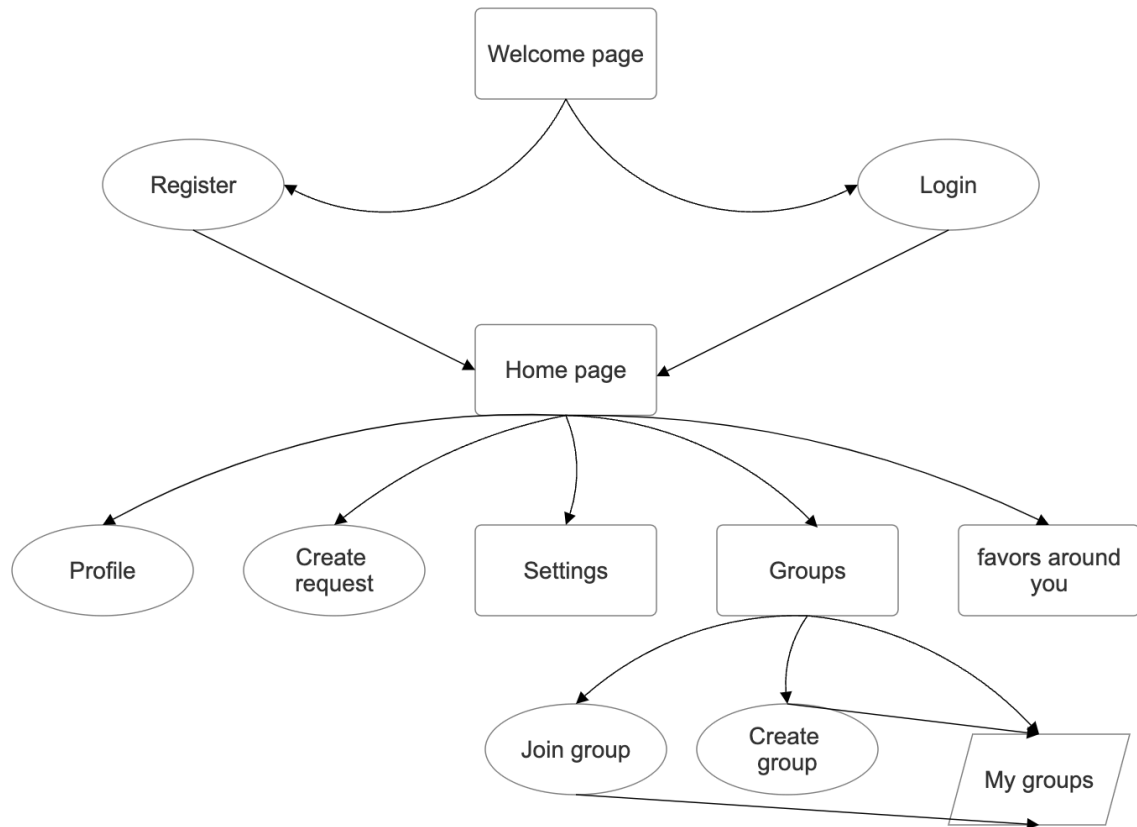
רשימת מסכים:

1. מסך הבית (home page) -  
תיאור: מציג מידע כללי וקישורים לתכנים עיקריים.  
רכיבים: כפתורי מעבר וגישה לתכני המערכת.
2. מסך כניסה (login page):  
תיאור: מאפשר למשתמשים קיימים להכנס למערכת.
3. מסך הרשמה (Register page):  
תיאור: מאפשר למשתמשים חדשים להרשם למערכת.
4. מסך פרופיל משתמש:  
תיאור: מציג את פרטי המשתמש, ומאפשר עדכון פרטים אישיים.
5. מסך הגדרות:  
תיאור: ניהול הגדרות אישיות והגדרות מערכת.
6. מסך תוצאות חיפוש:  
תיאור: לאחר בחירת מיקום לביצוע הטובה, יפתח למשתמש מסך המציג את כל הבקשות שנפתחו במיקום הנבחר.
7. מסך פתיחת בקשה לטובה :  
תיאור: מסך בו המשתמש יכול לפתוח בקשה, בין היתר יציין מיקום מדויק ותיאור הפעולה המבוקשת.
8. מסך אודות האפליקציה:  
תיאור: מסך שמכיל את כל הפרטים אודות האפליקציה, לרבות הצהרות משפטיות וכדומה.
9. מסך יצירת קבוצה:  
תיאור: מסמך המאפשר לך ליצור קבוצה שבה יהיה יכולות לעשות טובות לחברים ספציפיים
10. כניסה לקבוצה:  
תיאור: כניסה לקבוצה באמצעות קוד שנמצא אצל מנהל הקבוצה
11. התחברות לקבוצה ספציפית  
תיאור: התחברות לקבוצה ספציפית לצורך קבלת טובות
12. עדכון סוג משתמש  
תיאור: עדכון סוג משתמש

## 12.רשימת PopUp

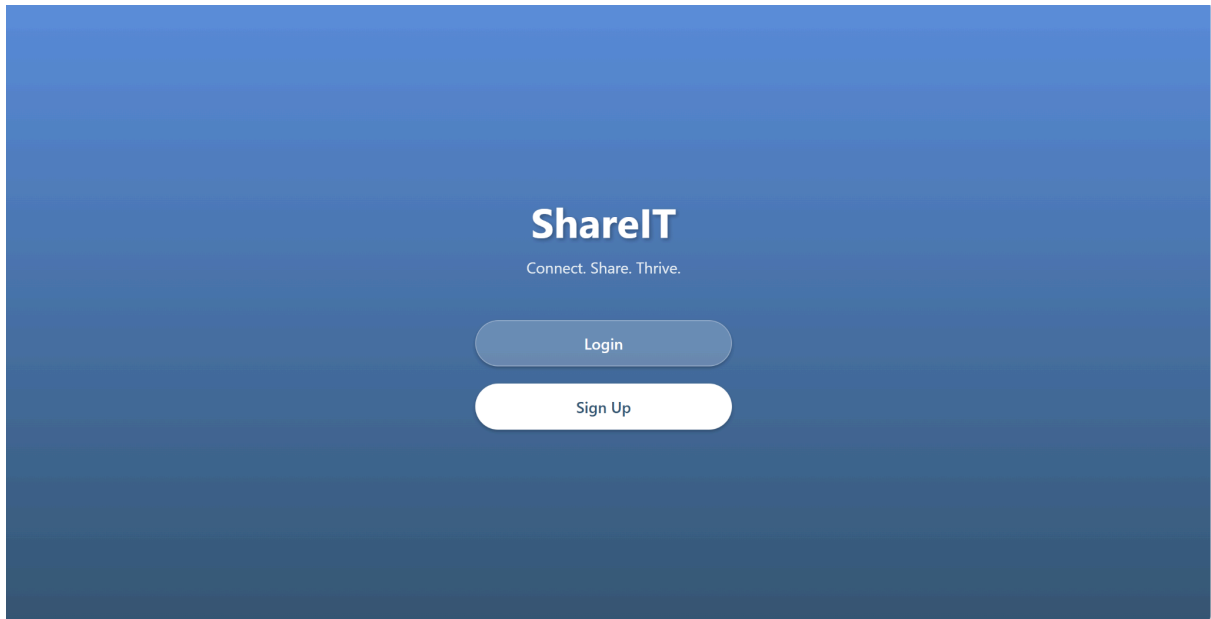
12.1 משימה פעילה : ברגע שהעלת טובה אז אחריי זה יפתח חלונית שלאחר עם צאט ופרטי המבצע

## Connection between screens



## Visual representation

welcome page:



### Welcome Back

Sign in to continue

Sign In

or continue with



## Sign Up

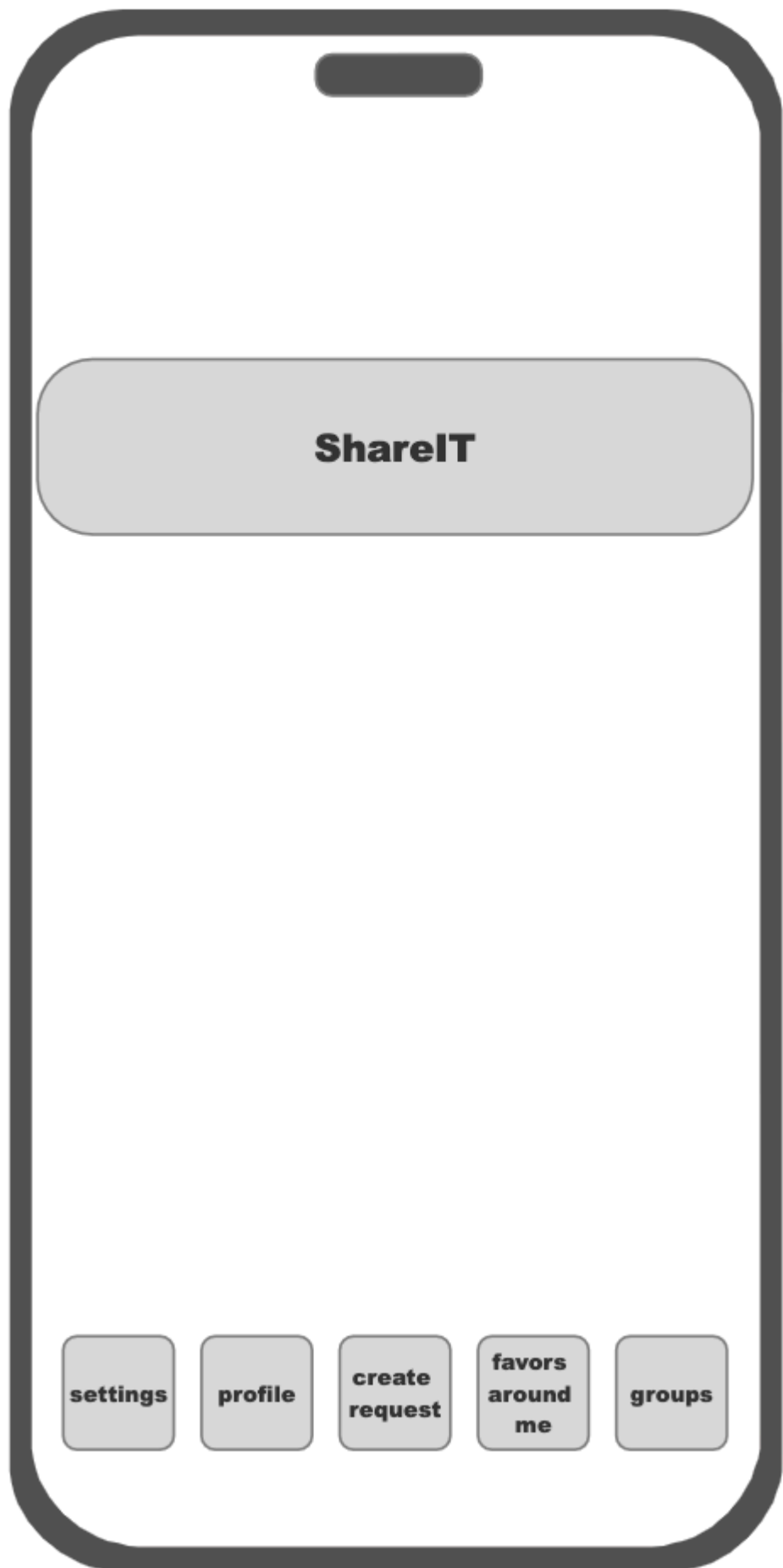
☐ I agree to the terms and conditions

Sign Up

Or continue with

Sign In with Google

Sign In with Facebook



**ShareIT**

**settings**

**profile**

**create  
request**

**favors  
around  
me**

**groups**



## Settings

**settings**

**settings**

**settings**

**settings**

**settings**

**settings**



## Free Text Search





## **My groups**

**Group 1**

**Group 2**

**Group 3**

**Group 4**



## PROFILE



### Account's information -

**mail:**  
**example@gmail.com**

**userName:**  
**israelisraeli1234**

**Change profile**



## Create request

**Instructions of  
how to create a  
request**

**Select location**

**Free text**

**Cancel  
request**

**Send  
request**

## החשיבות של MVVM בפרויקט שלנו

הפרויקט שלנו כולל מערכת מורכבת עם מסכים רבים, תהליכים דינמיים ואינטראקציות מגוונות בין משתמשים, כמו חיפוש, ניהול קבוצות ועדכון נתונים בזמן אמת. הבחירה ארכיטקטורת **MVVM** (Model-View-ViewModel) לפרויקט הזה היא הבחירה הנכונה ביותר עבורנו, מכיוון שהיא מאפשרת לנו לנהל את כל התהליכים בצורה מודולרית, גמישה וברורה, תוך שמירה על קוד איכותי ונגיש לתחזוקה.

### למה בחרנו ב-MVVM?

#### 1. ריבוי מסכים ותהליכים עצמאיים

בפרויקט שלנו ישנם 12 מסכים עיקריים, וכל מסך מביא איתו לוגיקה ייחודית ואתגרים משלו:

- **מסך הבית:** מציג תכנים עיקריים, גישה לטובות סביב המשתמש, וקישורים למסכים אחרים.
- **תוצאות חיפוש:** מציגות בקשות לטובות בהתאם למיקום שנבחר.
- **יצירת קבוצה וכניסה לקבוצה:** מטפלים בניהול קבוצות, חברים, ותהליכי יצירה וכניסה לקבוצות.

בחרנו ב-MVVM כי היא מאפשרת לנו לנהל כל מסך בנפרד באמצעות **ViewModel ייעודי**, שמטפל בלוגיקה העסקית של אותו מסך. לדוגמה:

- **Home View Model:** מטפל בכל המידע שמוצג בדף הבית, כולל קישורים וטובות קרובות.
- **Search Results View Model:** מטפל בשליפת נתוני חיפוש ועדכון תוצאות בזמן אמת.
- **Group View Model:** אחראי לניהול קבוצות, כולל יצירה, עדכון וכניסה לקבוצה.

השימוש ב-ViewModel ייעודי מבטיח שכל מסך נשאר עצמאי בתפקודו, מה שמקל על תחזוקה עתידית והרחבות.

#### 2. עדכונים בזמן אמת במסכים דינמיים

אחד האתגרים המרכזיים שלנו הוא ניהול עדכונים שוטפים בזמן אמת במסכים חשובים:

- **תוצאות חיפוש:** מציגות בקשות שמתעדכנות ברגע שמישהו מעלה בקשה חדשה.
- **משימה פעילה (PopUp):** מציג סטטוס משימה, צ'אט בין משתמשים, והתקדמות בזמן אמת.

MVVM נותנת לנו את הכלים הנדרשים לנהל עדכונים כאלה בצורה אוטומטית, באמצעות **LiveData** או **StateFlow**. לדוגמה:

- כאשר בקשה חדשה מתווספת באזור מסוים, מסך "תוצאות חיפוש" מתעדכן באופן מיידי.
- חלון ה-PopUp מתעדכן בכל פעם שמתנה סטטוס המשימה או שהמשתמש מקבל הודעה חדשה.

הכל מתבצע מבלי לכתוב קוד מיותר לעדכון ה-UI, מה שחוסך לנו זמן פיתוח ומפשט את התהליך.

### 3. הפרדה ברורה בין אחריות התצוגה ללוגיקה

אחד היתרונות המשמעותיים של MVVM הוא הפרדה ברורה בין הלוגיקה העסקית לתצוגה. בפרויקט שלנו ישנם תהליכים שדורשים **לוגיקה מורכבת**, כמו פה אינטראקטיבית שמראה את הטובות ה"פתוחות" בסביבתך, משימה זו דורשת גילוי מיקום המשתמש, הוצאה של מידע רב מהDB, עדכון על "לקיחת" טובה במידת הצורך ועוד.

MVVM מאפשרת לנו:

- **לרכז את הלוגיקה העסקית ב-ViewModel** – מה שהופך את הקוד שלנו למודולרי וקל לתחזוקה.
- **להשאיר את ה-View נקי** מכל לוגיקה עסקית, כך שהוא מתמקד רק בתצוגת המידע ואינטראקציות משתמש.

---

### 4. הרחבה עתידית וגמישות

בפרויקט כמו שלנו, שבו יש ניהול קבוצות, בקשות לטובות, וסוגי משתמשים, אנו יודעים שתמיד ייתכנו פיצ'רים חדשים שנרצה להוסיף. MVVM מאפשרת לנו להרחיב את המערכת בקלות:

- לדוגמה, אם נרצה להוסיף מנגנון התראות או מערכת דירוג למשתמשים, נוכל פשוט להוסיף ViewModel חדש לניהול הפיצ'ר, מבלי לפגוע בקוד הקיים.

---

### 5. ניהול נתונים מורכב

במסכים כמו "פרופיל משתמש", "תוצאות חיפוש", ו"עדכון סוג משתמש", נדרש שילוב של נתונים ממקורות שונים (שרת, מסד נתונים מקומי).

- MVVM מאפשרת לנו **לרכז את כל ניהול הנתונים ב-Model**, ולהשתמש ב-ViewModel כמתווך שמביא את הנתונים ל-View באופן מסודר.
- עם **Data Binding**, הנתונים מחוברים ישירות ל-UI, כך שה-View מתעדכן אוטומטית כשהנתונים משתנים.

---

### 6. שיפור חוויית המשתמש

השימוש ב-MVVM מבטיח חוויית משתמש חלקה ודינמית:

- המסכים מגיבים במהירות לכל שינוי בנתונים.
- אין צורך בטעינות מיותרות או עדכוני UI ידניים.
- המשתמש מקבל ממשק שמתפקד בצורה מהירה, ברורה ואינטואיטיבית.

---

איך MVVM מתבטא בפרויקט שלנו

## מסך הבית (Home Page):

- **Home ViewModel** מנהל את הנתונים המוצגים בדף הבית, כולל טובות קרובות וקישורים למסכים אחרים.
- השימוש ב-LiveData מבטיח שהמסך מתעדכן בכל פעם שמתווספת טובה חדשה.

## מסך תוצאות חיפוש:

- **Search Results ViewModel** מבצע את קריאות החיפוש לשרת ומחזיר את התוצאות.
- ה-View נשאר נקי, ומציג את הנתונים שמגיעים מה-ViewModel בלבד.

## מסך פרופיל משתמש:

- **Profile View Model** אחראי לשליפת נתוני המשתמש, עדכון פרטים חדשים, ולוגיקת ולידציה.

## מסך יצירת קבוצה:

- ה-ViewModel מטפל בכל הלוגיקה של יצירת קבוצה, כולל שם הקבוצה, קוד כניסה, והוספת חברים.

## משימה פעילה (PopUp):

- **LiveData** או **StateFlow** דואגים לעדכן את נתוני הצ'אט והמשימה בזמן אמת, כך שהמשתמש רואה את כל המידע המעודכן מיידית.

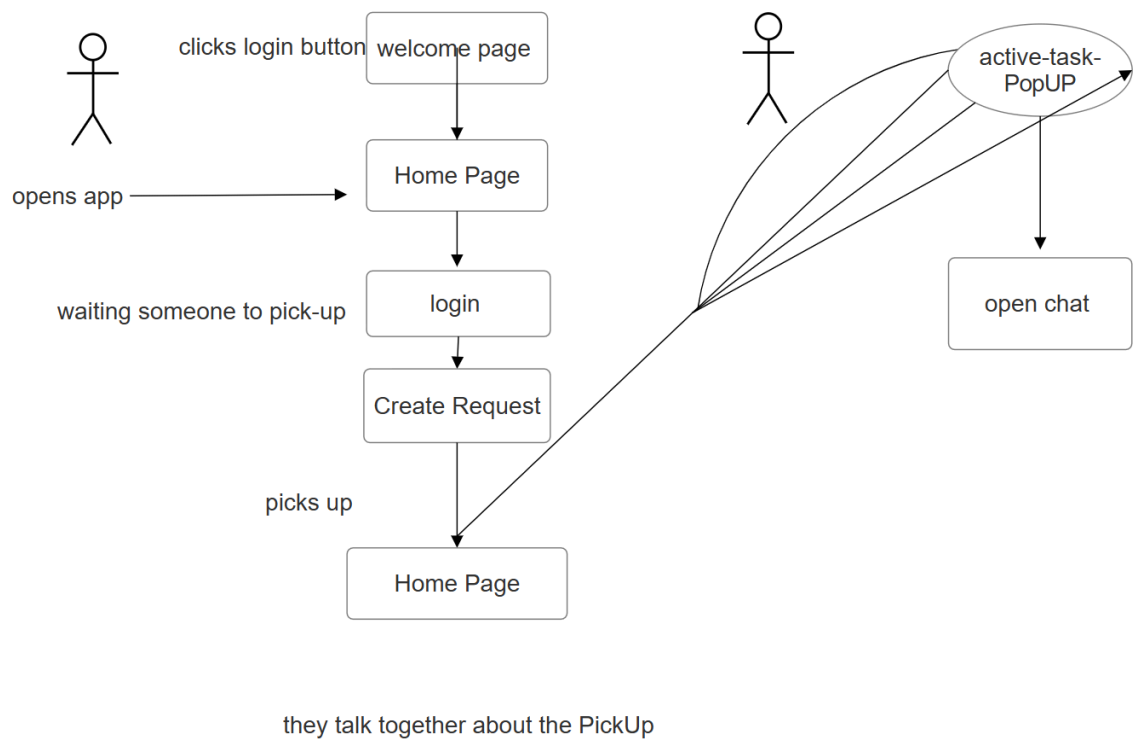
---

## סיכום – למה MVVM קריטי לפרויקט שלנו

1. **ניהול מסכים בצורה מודולרית:** כל מסך מנוהל כיחידה עצמאית, מה שמפשט תחזוקה ושינויים.
2. **עדכונים בזמן אמת:** נתונים מתעדכנים באופן אוטומטי ב-UI, בלי צורך בהתערבות ידנית.
3. **קוד נקי ומסודר:** הלוגיקה העסקית מופרדת מתצוגה, מה שמפחית טעויות ומקל על הבדיקות.
4. **יכולת הרחבה קלה:** הוספת פיצ'רים חדשים או מסכים נוספים לא משפיעה על המערכת הקיימת.
5. **שיפור חוויית המשתמש:** הממשק מגיב מהר ומשתמש בטכנולוגיות מתקדמות לעדכונים דינמיים.

בחרנו ב-MVVM כי אנחנו רוצים ליצור אפליקציה איכותית, גמישה ונוחה – הן למשתמשים והן למפתחים שיתחזקו את המערכת בעתיד.

## Sequence Diagram



## ERD (Entity-Relationship Diagram)

<b><u>Users</u></b>
user_id (pk)
email
username
password
access_level

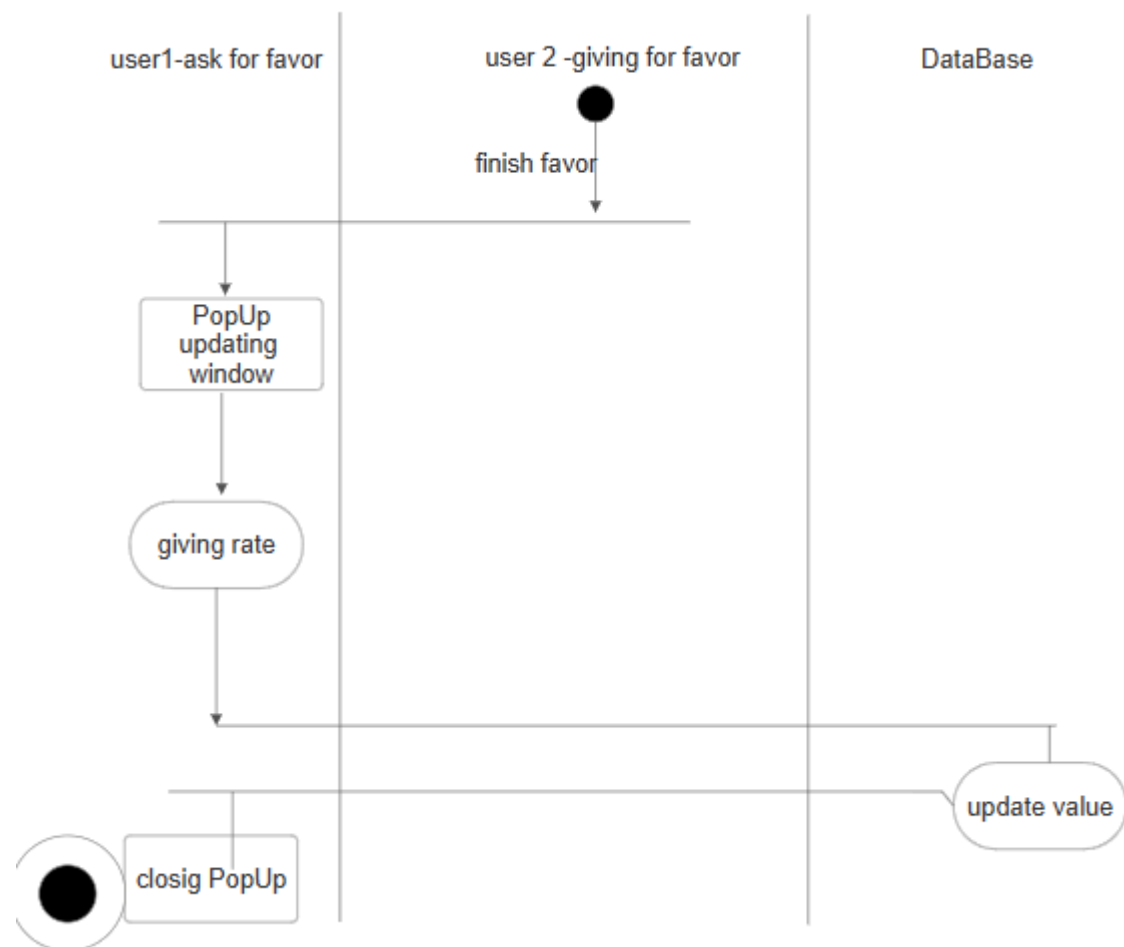
<b><u>Open Requests</u></b>
request_id (pk)
user_id (fk)
group_id (FK)
request_open_time
request_close_time
status
location

<b><u>Groups</u></b>
group_id (pk)
group_name
description

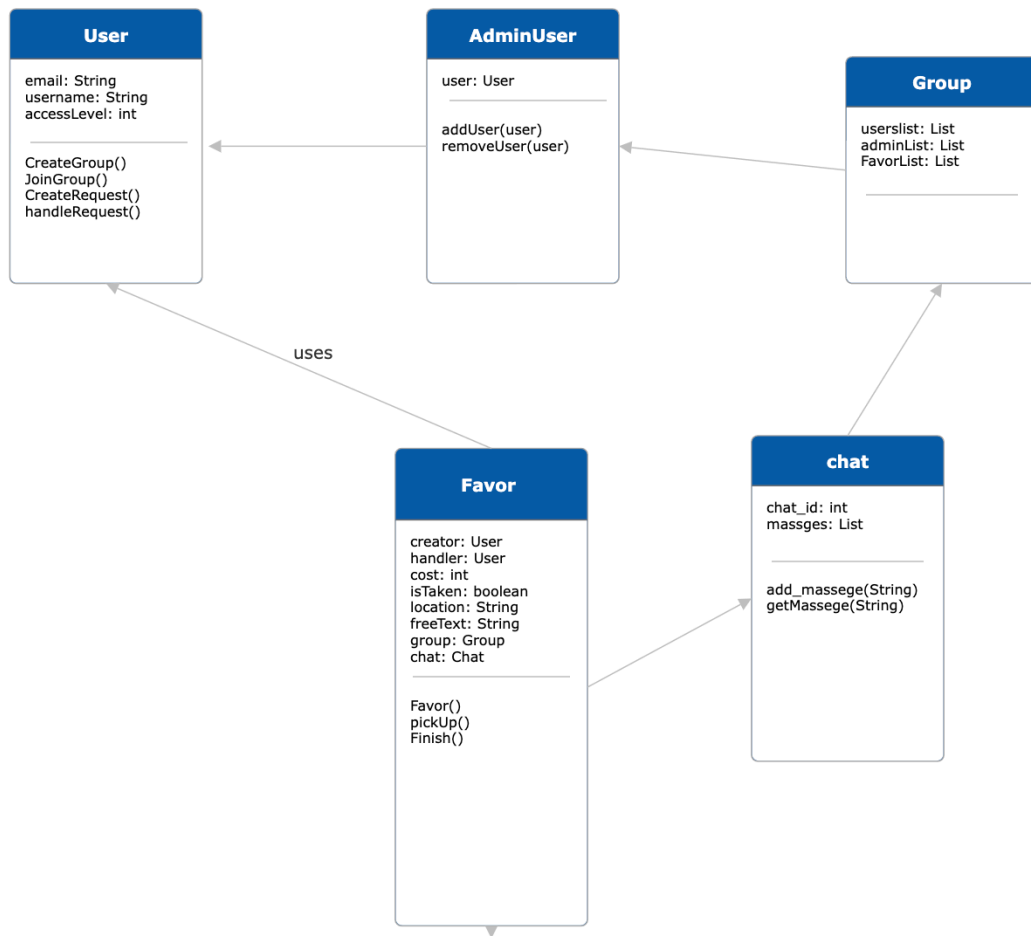
<b><u>user_group</u></b>
user_group_id (PK)
user_id (FK)
group_id (FK)
role
date_joined

## **Activity Diagram**

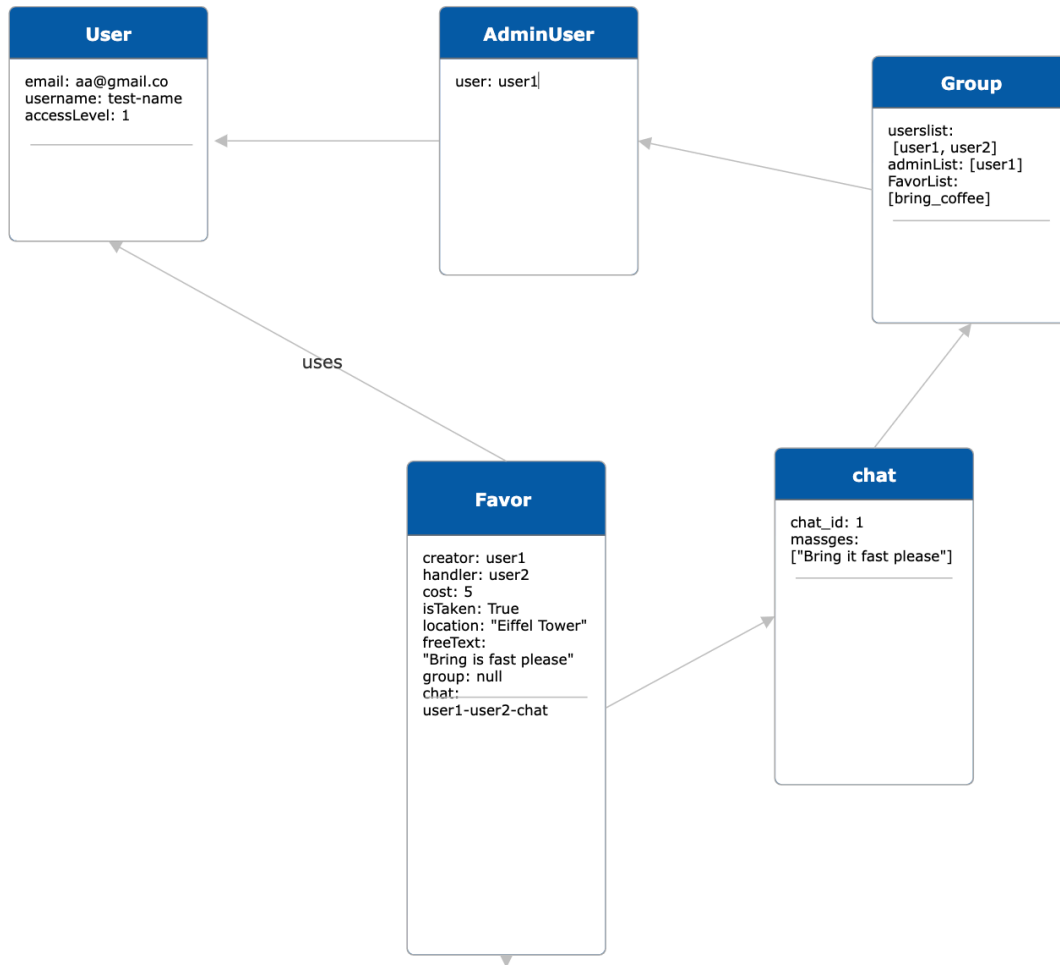




## Class diagram



## Object diagram



## State Machine Diagram

