

קבוצה 43

Join Me

Software Design Document

שחר מכלוף, תבור לוין וכן שטינמץ

8/01/2023

1. INTRODUCTION:

1.1 מטרה:

מסמך SDD זה מתאר את הארכיטקטורה ועיצוב המערכת של האפליקציה שלנו - Join Me.

1.2 Scope:

הפרויקט כולל איתור אנשים על מנת שהפעילות תוכל להתקיים אך לא כולל דאגה למיקום הפעילות או לחפצים הנדרשים לביצוע הפעילות.

1.3 Overview:

במסמך זה אנו מתארות את מבנה המערכת, תבנית העיצוב שהשתמשנו ואיך המערכת שלנו עונה לדרישות.

2. SYSTEM OVERVIEW:

מטרת העל של הפרויקט היא איתור קבוצת אנשים הרוצים לבצע פעילות משותפת עם אנשים אחרים על בסיס מיקום וזמינות (לדוגמה: משחק כדורגל).

האפליקציה מכילה 2 הרשאות: משתמש ומנהל.

המשתמש יכול ליצור קבוצות, להצטרף לקבוצות קיימות, לדווח על משתמשים ועוד.

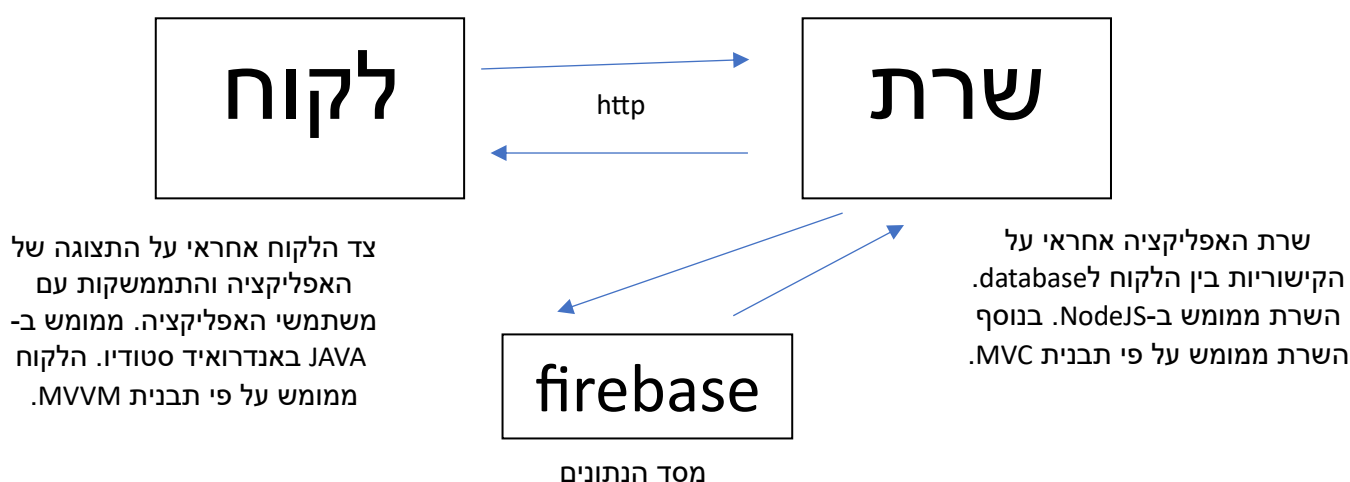
המנהל יכול לחסום משתמשים, ליצור סטטיסטיקות ועוד.

האפליקציה ממומשת על ידי צד שרת וצד לקוח כך שכל הלוגיקה נמצאת בשרת והview נמצא בצד הלקוח.

גם השרת ממומש על ידי תבנית MVC והלקוח ממומש על ידי תבנית MVVM.

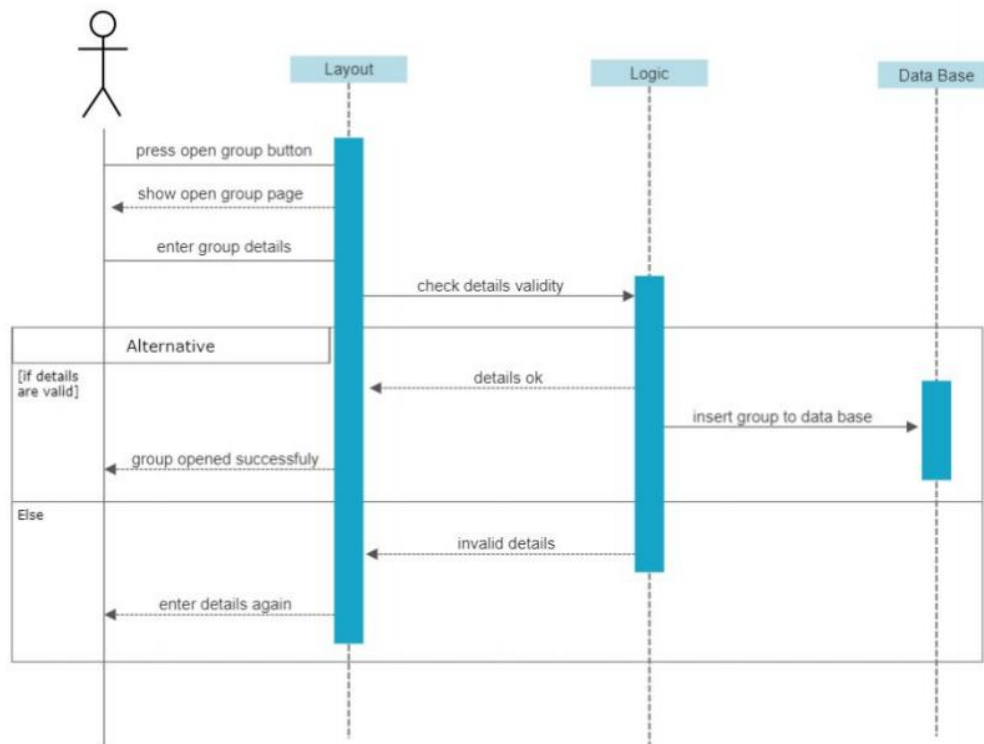
3. SYSTEM ARCHITECTURE:

3.1 Architectural Design



:Decomposition Description 3.2

Sequence diagram:



chen: Admin

name : Chen
mail : "jiayhb@gmail.com"

ur1 : UserRow

uid: 535a5v589b69h9m9m67
name: Raz
mail: "raz4447@gmail.com"

g1 : Group

title : Football
city : Ariel
time : 18:00
date : 25/12/2022
head_of_group_uid : 535a5v589b69h9m9m67
min_participants : 2
max_participants : 7
num_of_participant : 15
is_happened : yes
participants{535a5v589b69h9m9m67,63da5v509i69i9p9m67,
8a4ilv5l9b7859a9w60,544a5f289as1h9h3s75,737a5o783b64i9m9o12}

c1 : Contact

category: Football
location: Ariel
date: 25/12/2022
id : 5s5v8n4te3g6j4t2e

u1 : User

uid: 435i5v589i62h7m9m89
name: Raz
phone: 0526700186
mail: "raz4447@gmail.com"
birth_date: 01/01/1998
my_groups: {52s2g4hy5r3w9ws64g2r}
num_of_reports: 3
success_creating_groups: 1
groups_I_joined: NULL

3.3 Design Rationale

בחרנו בתבניות MVC ו-MVVM מכיוון שיש לנו כאן התעסקות עם תצוגה למשתמש וגם התעסקות עם דאטה בייס ולכן תבניות אלו היו נוחות לנו. ב-view אנו דואגות לתצוגת המשתמש, ב-model אנו דואגות ללוגיקה ולתקשורת עם השרת וב-view-model אנו מקשרות בין ה-view ל-model.

תבנית נוספת ששקלנו להשתמש בה היא תבנית השכבות. בחרנו לא להשתמש בה מכיוון שאין הרבה בדיקות לוגיקה באפליקציה שלנו ולכן העדפנו שהלוגיקה תהיה יחד עם ההתעסקות עם database.

4. DATA DESIGN

4.1 Data Description

הדאטה שלנו נשמר ב-firebase. קיימים 2 collections מרכזיים: 1. userById. 2. Groups.

קיים אובייקט המתאים לכל document.

4.2 Data Dictionary

User

```
[+] uid: String
[+] name: String
[+] phone: String
[+] mail: String
[+] birth_date: String
[+] my_groups: ArrayList<String>
[+] num_of_reports: int
[+] success_creating_groups: int
[+] groups_I_joined: ArrayList<String>
```

```
getUid()
getGroups_I_joined()
setGroups_I_joined(ArrayList<String> groups_I_joined)
String getName()
String getPhone()
String getMail()
ArrayList<String> getMy_groups()
int getNum_of_reports()
void setName(String name)
void setPhone(String phone)
void setMail(String mail)
void setMy_groups(ArrayList<String> my_groups)
void setNum_of_reports(int num_of_reports)
int getSuccess_creating_groups()
void setSuccess_creating_groups(int success_creating_groups)
String getBirth_date()
void setBirth_date(String birth_date)
void addGroup(String gid)
void addGroupIjoined(String gid)
```

Group

```
[+] title : String
[+] city : String
[+] time : String
[+] date : String
[+] head_of_group_uid : String
[+] min_participants : int
[+] max_participants : int
[+] num_of_participant : int
[+] is_happened : boolean
[+] participants: ArrayList<String>
```

```
getTitle()
int getNum_of_participant()
String getCity()
String getTime()
boolean isIs_happened()
public ArrayList<String> getParticipants()
void setTitle(String title)
void setNum_of_participant(int num_of_participant)
void setCity(String city)
void setTime(String time)
void setIs_happened(boolean is_happened)
void setParticipants(ArrayList<String> participants)
String getHead_of_group()
void setHead_of_group(String head_of_group)
int getMin_participants()
int getMax_participants()
void setMin_participants(int min_participants)
void setMax_participants(int max_participants)
String getDate()
void setDate(String date)
void addParticipant(String uid)
```

5. COMPONENT DESIGN:

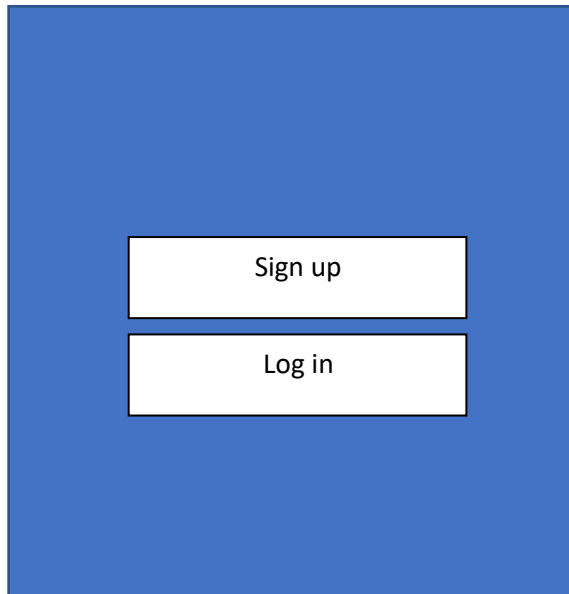
כל אובייקט מכיל Getters & Setters. האובייקטים שלנו מאוד פשוטים ולכן לא צריכים פונקציות נוספות.

6. HUMAN INTERFACE DESIGN:

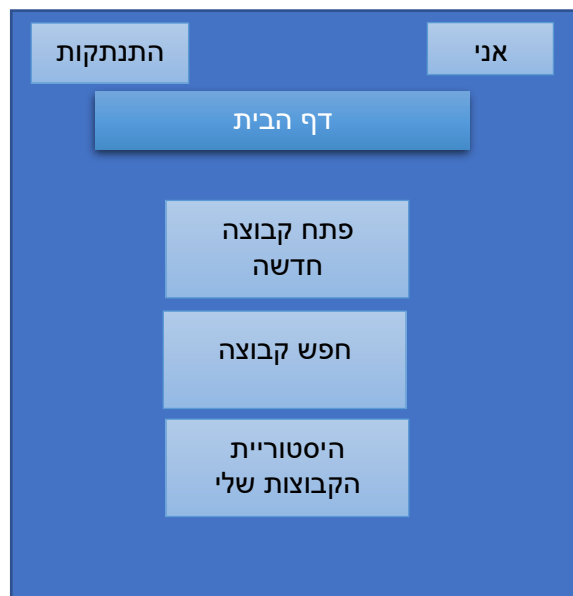
6.1 Overview of User Interface:

התצוגה שלנו תהיה מאוד ברורה ומובנת למשתמש ויהיה קל להבין אותה.

6.2 Screen Images:



מסך כניסה:



דף הבית

חזור

הרשמה

שם:

שנת לידה:

פלאפון:

מייל:

סיסמה:

אימות סיסמה:

אישור

מסך הרשמה:

חזור

התחברות

מייל:

סיסמה:

אישור

מסך התחברות:

חזור

התנתקות

פתיחת קבוצה

מטרת הקבוצה:

מיקום:

שעה:

מספר משתתפים נדרש:

צור קבוצה

פתיחת קבוצה

התנתקות

חזור

חיפוש קבוצה

כדורגל 17:00
מניין 14:00

חיפוש קבוצה:

7. REQUIREMENTS MATRIX

רכיב מערכת	דרישה
Create group activity	משתמש יכול לפתוח קבוצה בהתאם לצורך/לרצון שלו
Find group activity	משתמש יכול להצטרף לקבוצה בהתאם לצורך/לרצון שלו
History activity	משתמש יכול לראות את היסטוריית השתתפותו בקבוצות
Update details activity	משתמש יכול לערוך את הפרטים שלו
Block user activity	מנהל יכול להסיר משתמשים לא אמינים
Report user activity	משתמש יוכל לדווח על משתמש לא אמין
Statistics activity	מנהל יכול ליצור סטטיסטיקות הצלחת קבוצות ובהתאם לכך להגיע למסקנות
History activity	דיווח שקבוצה יצאה לפועל
Admin History activity	מנהל יכול לראות את היסטוריית כל הקבוצות שנפתחו
Log out button	התנתקות משתמש
Location service	המערכת תזהה משתמשים הנמצאים בקרבת מקום ותציע להם להצטרף לקבוצה
BigMI	המערכת תציע למשתמש להצטרף לקבוצות לפי היסטוריית ההעדפות שלו
Whatsapp	כאשר קבוצה מתגבשת המערכת תפתח למשתתפים קבוצת וואצאפ על מנת שיוכלו לתקשר