אוניברסיטת בן גוריון בנגב
מחלקה למדעי המחשב
מבני נתונים – תשע"ד סמסטר ב

# מבני נתונים – תרגיל 2 (מעשי)

**תאריך פרסום**: 23.03.2014
**תאריך הגשה**: 10.4.2014

**מרצה ומתרגלים אחראים**: דר' דקל צור, עדיאל אשרוב, אריאל הנמן.

## נושאי העבודה:

- תכנון של מבנה נתונים התומך ב- ADT(Abstract Data Type) שיוגדר בהמשך התרגיל.
- ביצוע סימולציה נכונה ויעילה של התהליך שיוגדר בהמשך התרגיל.

הערות ושאלות יש להפנות בפורום הקורס בכתובת הבאה:
http://www.cs.bgu.ac.il/~ds142/Forum

## רקע כללי:

Whatsapp הינה אפליקציית מסרים מידיים המאפשרת שליחת הודעות, הודעות קוליות ושיתוף תמונות בין המשתמשים שלה. Whatsapp שולחת הודעות בין המשתמשים באמצעות Wi-fi / אינטרנט אלחוטי או חבילת ה - Data של המשתמש. למעשה אין צורך בסים וחיבור ישיר לרשת סלולרית, כל מה שדרוש הוא חיבור אלחוטי לאינטרנט כדי לתקשר

לפרטים נוספים:

http://en.wikipedia.org/wiki/WhatsApp
http://www.whatsapp.com/

## מבוא:

בעבודה זו תממשו מבנה נתונים שתומך בפונקציונליות דומה לזו של Whatsapp .
הפעולות בהן מבנה הנתונים נדרש לתמוך בהן:

| שם פעולה | תיאור פעולה | זמן ריצה במקרה הגרוע ביותר |
|---|---|---|
| *AddUser* | הוספת משתמש חדש למערכת | $O(1)$ |
| *CreateGroup* | נוצרת קבוצת שיחה בין שני משתמשים או יותר | $O(numberOfUsers)$ |
| *AddMemberToGroup* | הוספת משתמש לקבוצת שיחה | $O(numberOfGroups + numberOfUsers * maxNumOfConversationInUser)$ |
| *SendMessage* | שליחת הודעה לקבוצת שיחה, הודעה זו מגיעה לכולם | $O(numberOfGroups + maxNumberOfUsersInGroup * maxNumOfConversationInUser)$ |
| *RemoveUserFromGroup* | הסרת משתמש מקבוצה | $O(numOfGroup + numOfUsers + maxNumOfConversationinUser)$ |
| *SearchMessageByText* | חיפוש טקסט ברצף הודעות של משתמש | $O(numberOfUsers + maxNumOfConversationInUser + maxNumOfMessagesInConversation)$ |

הבדל עיקרי בין האפליקציה האמיתית למימוש הנדרש הוא שניתן לתקשר רק באמצעות קבוצות (כלומר גם על מנת ששני משתמשים בלבד יוכלו לדבר, צריכה להיות קבוצה שכוללת את שניהם).

## מחלקות:

### *המחלקה Message:*

מחלקה זו מייצגת הודעה אחת. להודעה יש את שם השולח ותוכן ההודעה. למחלקה יש גם שדה next מסוג Message מכיוון שהודעה אחת יכולה להיות חלק מרשימת הודעות בשיחה.

### *המחלקה Conversation:*

המחלקה מייצגת שיחה. לשיחה יש את שם הקבוצה בה היא מתנהלת ורשימה משורשרת של הודעות (אובייקטים מסוג Message). למחלקה יש גם שדה next מסוג Conversation מכיוון שהיא יכולה להיות חלק מרשימת שיחות אצל המשתמש.

### *המחלקה User:*

המחלקה מייצגת משתמש. למשתמש יש שם ורשימה משורשרת של כל השיחות שלו (אובייקטים מסוג Conversation). רשימת השיחות מתנהלת באופן הבא: בכל פעם ששיחה מתעדכנת (נוספת לה הודעה) היא עולה לראש הרשימה. למחלקה זו אין שדה next ולצורך שרשור משתמשים ברשימה יש להיעזר במחלקה UserListNode.

### *המחלקה UserListNode:*

מחלקה זו היא מחלקת עזר. כדי ליצור רשימת User אזי יוצרים רשימה של UserListNode המתארת חוליה ברשימת משתמשים. כל אובייקט מחזיק אובייקט מסוג User ומצביע ל Next שהוא מסוג UserListNode. משתמשים בה מכיוון ש-User מסוימים יהיו חלק גם מרשימת המשתמשים הכללית וגם מרשימות משתמשים של קבוצות ואנו רוצים למנוע חזרה מיותרת על אובייקטים.

### *המחלקה Group:*

המחלקה מייצגת קבוצה. לקבוצה יש שם ורשימה משורשרת של המשתמשים שלה (אובייקטים מסוג UserListNode). כמו כן לקבוצה יש מצביע ל Next מסוג Group מכיוון שקבוצה יכולה להיות חלק מרשימת קבוצות.

### *המחלקה Whatsapp:*

המחלקה מייצגת את כל מערכת Whatsapp. היא מכילה רשימה משורשרת של משתמשים (כל המשתמשים) ורשימה משורשרת של קבוצות (כל הקבוצות במערכת) כאשר כל קבוצה מצביעה על משתמשים מתוך רשימת המשתמשים. גם כאן רשימת המשתמשים תהיה רשימה של UserListNode שכל אחד מצביע על User. (ראו דוגמה).

## המשימה:

עליכם לממש את המחלקה Whatsapp כך שתתמוך בשיטות הבאות:

Members:

```
UserListNode UsersHead;//Pointer to the head of the users list
Group GroupsHead;//Pointer to the head of the groups list
```

Methods:

```
public void AddUser(String inUserName);
```

מתודה זו יוצרת משתמש חדש בשם inUserName ומוסיפה אותו לרשימת המשתמשים שיש לנו במערכת. משתמש חדש מתווסף לסוף הרשימה. פעולה זו צריכה להתבצע ב - $O(1)$.

```
public void CreateGroup(String inInitiator,String inGroupName);
```

מתודה זו יוצרת קבוצה חדשה של משתמשים ומוסיפה אותה לרשימת הקבוצות (קבוצה חדשה מתווספת בסוף הרשימה). היא מקבלת כפרמטר את המשתמש אשר יזם את הקבוצה וכפרמטר נוסף את שם הקבוצה שנוצרת. רשימת המשתמשים של קבוצה שהרגע נוצרה תכיל משתמש אחד, מי שיצר אותה. מתודה זו צריכה להתבצע במגבלת זמן הריצה של $O(numberOfUsers)$ , כלומר מספר המשתמשים.

```
public void AddMemberToGroup(String invitorUser,String invitedUser,String
inWelcmeText,String inGroupName);
```

מתודה זו מוסיפה משתמש לקבוצה. משתמש חדש מתווסף לסוף רשימת המשתמשים בקבוצה. המתודה מקבלת כפרמטר את שם המשתמש שיוזם את הצירוף (כל משתמש שכרגע בקבוצה יכול ליזום את הצירוף), שם המשתמש שצריך להוסיף, הודעת "ברוך הבא" שתפורסם בקבוצה ואת שם הקבוצה. הודעת "ברוך הבא" תתווסף באותו רגע לכל השיחות של הקבוצה הזאת במערכת.

למשל אם יש קבוצה עם שני משתמשים X ו Y, לכל משתמש יש שיחה של הקבוצה ברשימת השיחות שלו. כש -X יצרף גם את Z, תיווצר שיחה חדשה של הקבוצה אצל Z ואז לכל המשתמשים תופיע הודעה חדשה בשיחה של הקבוצה. ההודעה תכלול את הודעת "ברוך הבא" והשולח יהיה X מכיוון שהוא זה שצירף לקבוצה. **<u>השיחה של אותה קבוצה אצל כל משתמשי הקבוצה צריכה לעלות לראש רשימת השיחות מכיוון שהרגע התקבלה אצלן הודעה.</u>** המתודה צריכה לרוץ במגבלת הזמן ריצה של $O(numberOfGroups + numberOfUsers * maxNumOfConversationInUser)$ כלומר מספר הקבוצות ועוד מספר המשתמשים כפול מספר הקבוצות/שיחות.

```
public void SendMessage(String inSender,String inText,String inGroupName);
```

מתודה זו שולחת הודעה לחברי הקבוצה. היא מקבלת כפרמטרים את שם השולח, הטקסט שהוא רוצה לשלוח לקבוצה ואת שם הקבוצה. לאחר פעולה זו תופיע הודעה חדשה בשיחות של הקבוצה אצל כל חברי הקבוצה (כולל השולח) עם הטקסט ושם השולח. הודעה חדשה מופיעה בסוף רשימת ההודעות בשיחה. **<u>השיחה של אותה קבוצה אצל כל משתמשי הקבוצה צריכה לעלות לראש רשימת השיחות מכיוון שהרגע התקבלה אצלן הודעה.</u>**

פעולה זו צריכה להתבצע ב - $O(numberOfGroups + maxNumberOfUsersInGroup * maxNumOfConversationInUser)$ , כלומר מספר הקבוצות ועוד מספר מקסימלי של משתמשים בקבוצה כפול מספר הקבוצות/שיחות.

```
public void RemoveUserFromGroup(String inUserName,String inGroupName);
```

מתודה זו מוחקת את המשתמש מהקבוצה (כלומר הקבוצה כבר לא תצביע למשתמש הזה). היא מקבלת כפרמטרים את שם המשתמש ואת שם הקבוצה. לאחר פעולה זו תמחק השיחה של הקבוצה הזאת מרשימת השיחות של המשתמש שעזב.

במידה והקבוצה ריקה לאחר העזיבה, הקבוצה תמחק מרשימת הקבוצות. פעולה זו צריכה להתבצע ב $O(numOfGroup + numOfUsers + maxNumOfConversationInUser)$, כלומר מספר הקבוצות ועוד מספר המשתמשים ועוד מספר השיחות אצל המשתמש בעל הכי הרבה שיחות.

```
public Message[] SearchMessageByText(String inUserName,String
inText,String inGroupName);
```

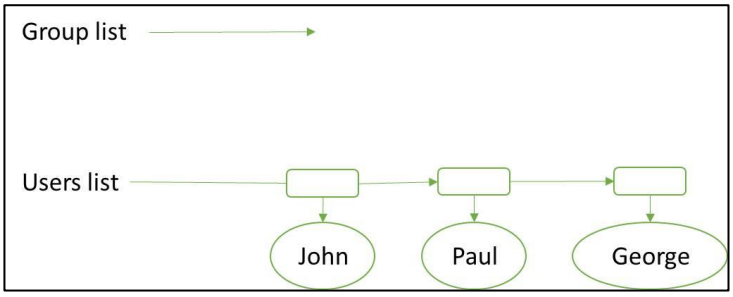
מתודה זו מקבלת כפרמטרים שם משתמש, טקסט לחיפוש ושם קבוצה. המתודה צריכה לחפש בשיחת הקבוצה אצל המשתמש, הודעות שהטקסט שלהן מכיל את הטקסט לחיפוש ומחזירה מערך של כל ההודעות האלה. המערך יהיה מסודר כך שהודעות ישנות תהיינה לפני הודעות חדשות. ניתן להניח שחיפוש טקסט בהודעה אחת מתבצע ב $O(1)$. פעולה זו צריכה להתבצע ב-

$O(numberOfUsers + maxNumOfConversationInUser + maxNumOfMessagesInConversation)$,
כלומר מספר המשתמשים ועוד מספר השיחות אצל המשתמש בעל הכי הרבה שיחות ועוד מספר ההודעות בשיחה בעלת מספר ההודעות המקסימלי.

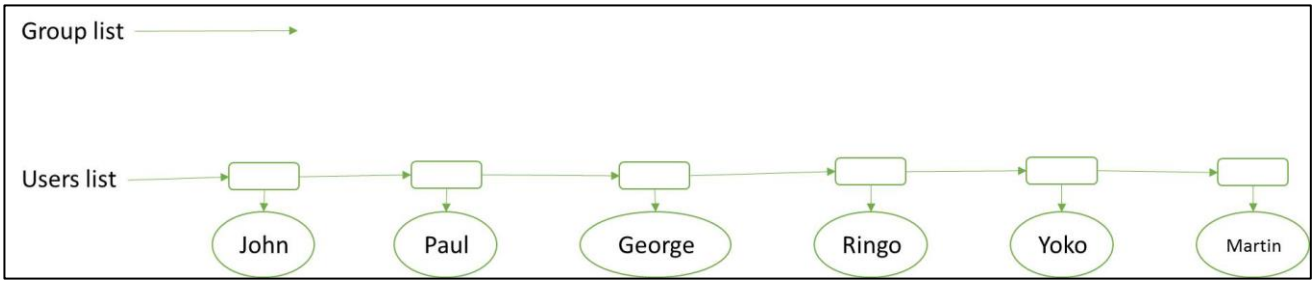
## דוגמה:

```
Start:
```

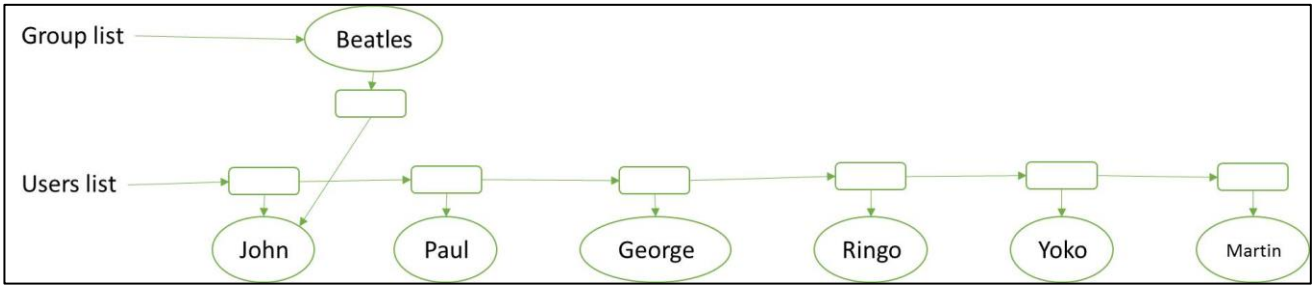

```
AddUser("John");
AddUser("Paul");
AddUser("George");
```



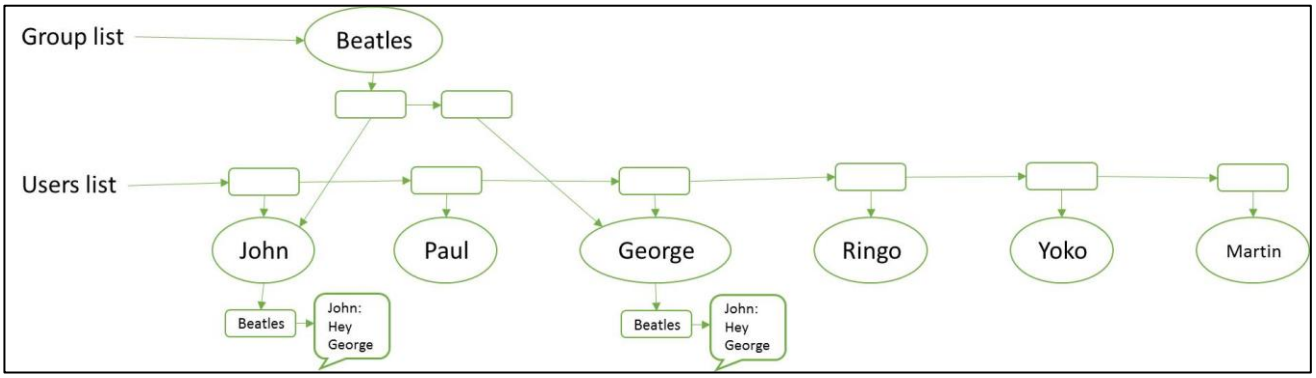

```
AddUser("Ringo");
AddUser("Yoko");
AddUser("Martin");
```



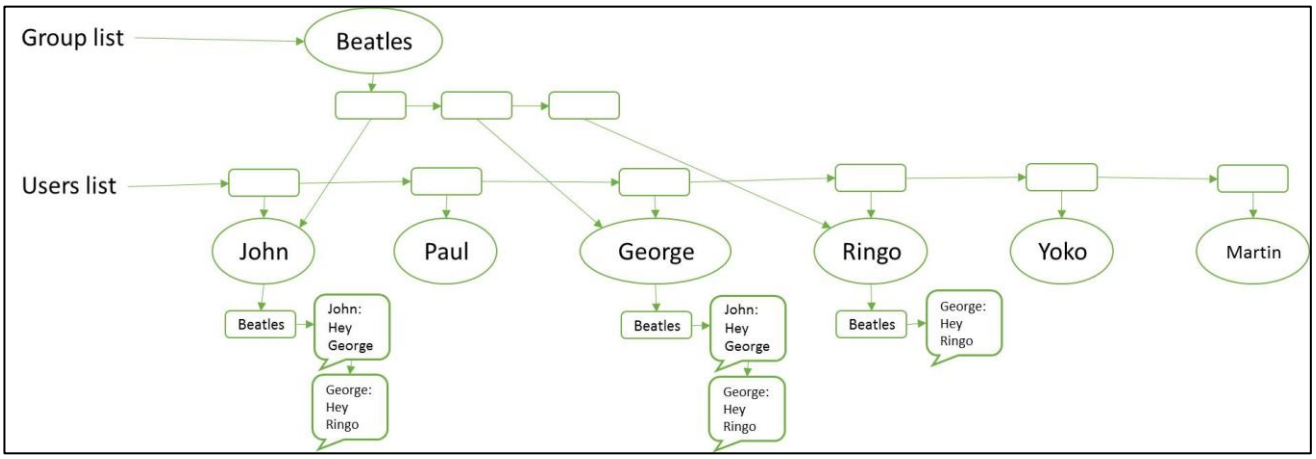

```
CreateGroup("John","Beatles");
```



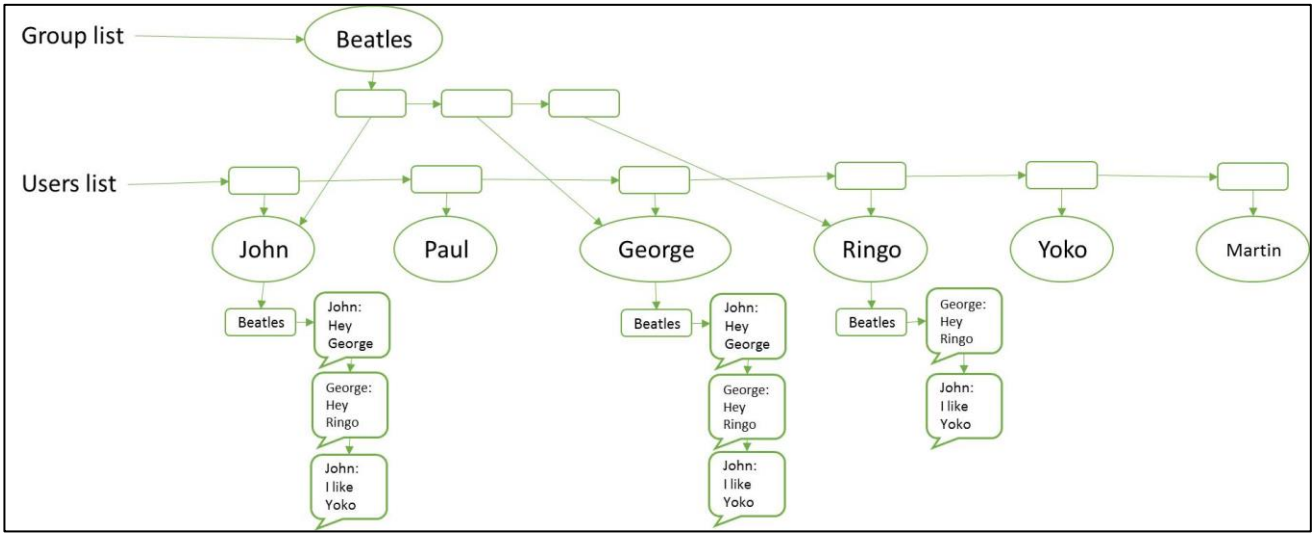

```
AddMemberToGroup("John","George","Hey George","Beatles");
```



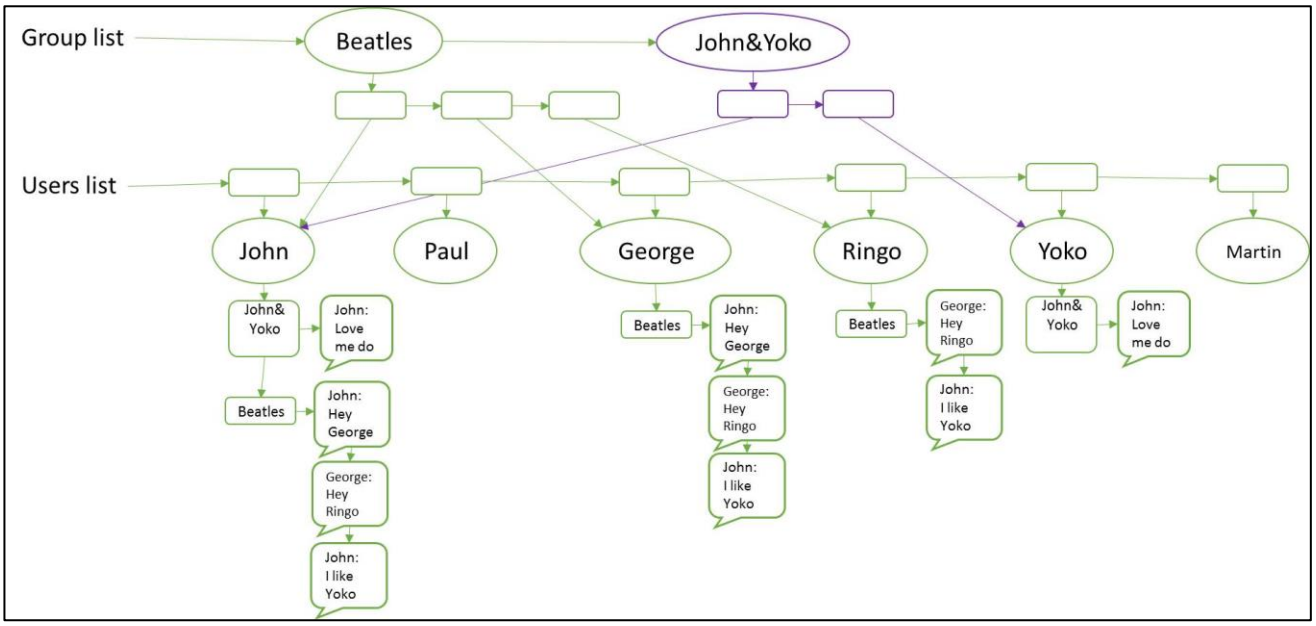

```
AddMemberToGroup("George","Ringo","Hey Ringo","Beatles");
```




```
SendMessage("John","I like Yoko","Beatles");
```

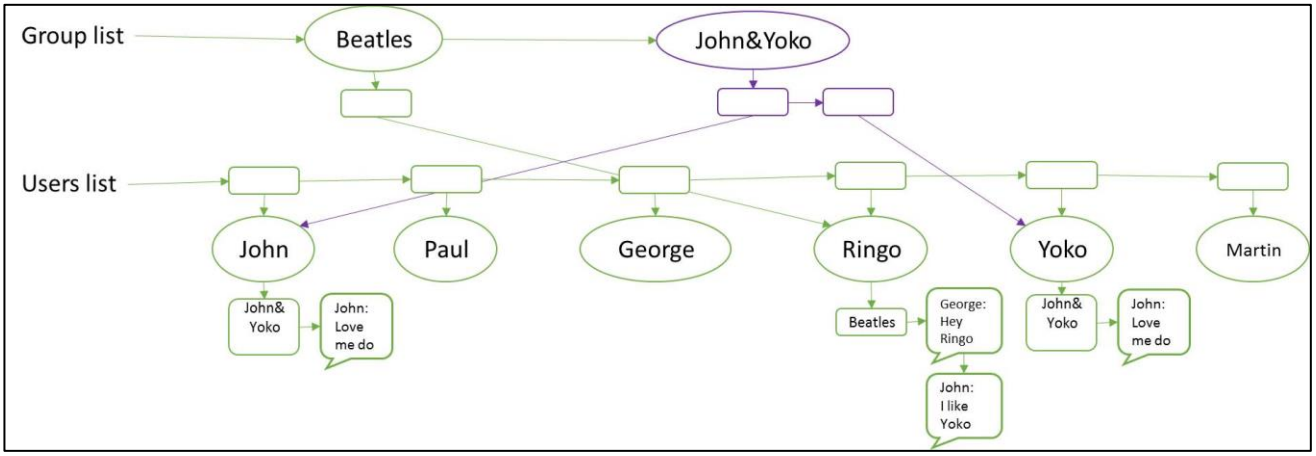



```
CreateGroup("John","John&Yoko");
AddMemberToGroup("John","Yoko","Love me do","John&Yoko");
```

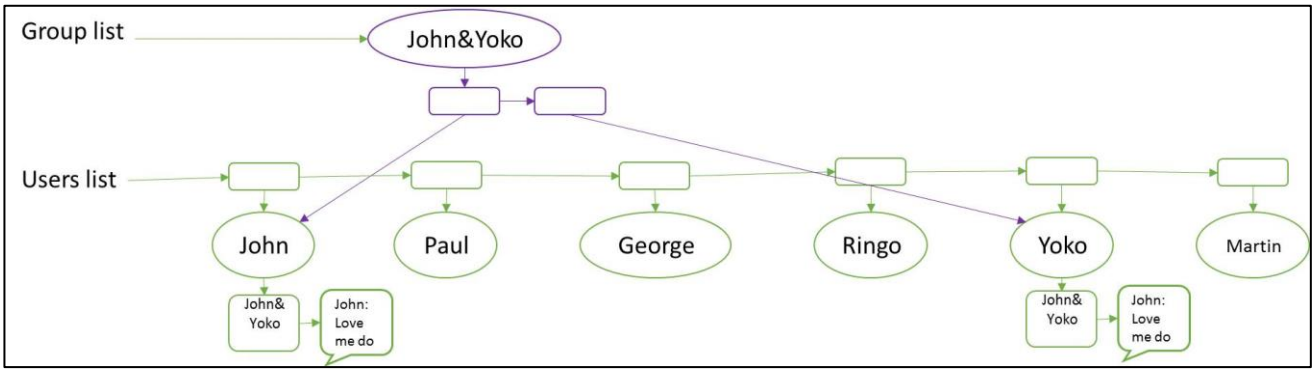



\\Notice that John&Yoko conversation is now in the head of John's list

```
RemoveUserFromGroup("John","Beatles");
RemoveUserFromGroup("George","Beatles");
```




```
RemoveUserFromGroup("Ringo","Beatles");
```




\\Notice that The Group Beatles was deleted from the Group.

## הערות חשובות ודרישות הגשה:

1. אין להשתמש במבנה נתונים גנריים הקיימים ב - Java במימוש העבודה. עליכם לממש את הרשימות בעצמכם.
2. ניתן להניח שהקלט יהיה תקין. לא תתבקשו להוסיף משתמש שכבר קיים או קבוצה שכבר קיימת, לא תתבקשו לשלוח הודעה לקבוצה שלא קיימת או ע"י משתמש לא קיים וכו'. יתכן שתתבקשו לחפש טקסט שלא מופיע בשיחה ואז יש להחזיר null.
3. אתם מקבלים את קבצי המחלקות ממומשים חלקית. ניתן להוסיף להם מתודות, בנאים, ושדות .
4. במחלקה Whatsapp אתם מקבלים את חתימות הפונקציות שתצטרכו לממש כחלק מהעבודה.
5. **בכל המחלקות אתם מקבלים פונקציות (כבר כתובות) שמחזירות String. הפונקציות האלה הן למטרת בדיקה (גם שלכם וגם שלנו) יש להשאיר את כל הפונקציות שקיבלתם. יש להשתמש בכל המשתנים שכבר קיבלתם.**
6. בין הקבצים, תקבלו גם קובץ Main.java שישמש אתכם לבדיקה. לאחר שתסיימו את התכנית אתם יכולים להריץ את ה- main באותה מחלקה ולהשוות את תוצאות ההדפסה עם הפלט שנפרסם באתר כדי לדעת אם התכנית עובדת כמו שצריך.
7. את העבודה יש להגיש ל Submission system.
8. עליכם להגיש קובץ zip בשם assignment2.zip המכיל בתוכו תיקיית src ובה קבצי הג'אווה של העבודה (בלי קובץ Main).
9. סביבת העבודה בה תיבדקנה העבודות הינה JavaSE-1.6/7
10. עליכם לדאוג כי עבודותיכם יתקמפלו וירוצו בסביבת eclipse תחת גרסאות Java הנזכרות לעיל. עבודות שלא יתקמפלו – לא יבדקו.
11. עבודותיכם יבדקו באמצעות כלי בדיקה אוטומטים הבודקים קורלציה בין עבודות. נא לא להעתיק! להזכירכם, המחלקה רואה בחומרה רבה העתקות.

12. באתר הקורס בדף התרגיל יהיו קבצי הג'אווה וקובץ פלט לדוגמא.

# בהצלחה!