

# אפיון פונקציונליות

**כותרת:** רשת חברתית מוטמעת בדפי אינטרנט.

**תיאור:** המערכת הינה תוסף שניתן להוסיף לכל דף אינטרנט ואשר תהפוך כל דף אינטרנט לרשת חברתית שהנושא שלה הוא התוכן המוצג בדף. המערכת תעשה זאת על ידי כך שתציג כפתור בצד הדף, שבלחיצה עליו יפתח ממשק ובו האפשרויות הבאות:

**פורום:** אפשרות פתיחת אשכול ושרשור תגובות.

**צ'אט:** אפשרות כתיבת הודעה .

**קבצים:** אפשרות להעלות ולשתף קבצים.

**עוקב:** אפשרות לעקוב אחר משתמש שכותב בדיון כלשהוא.

**חברות:** אפשרות להיות חבר של משתמש .

**דיווח:** אפשרות דיווח תוכן פוגעני.

**דירוג:** אפשרות לדרג תגובות פורום וקבצים שעלו לאתר .

כל התכנים שיתווספו באפשרויות הללו יהיו ספציפיים לדף אליו הם שויכו, כך שהמעבר בין הדפים יחשוף תכנים שונים

## use case

**חלון המערכת:** אפשרות לגלות/הסתיר בעזרת כפתורי ניווט ואפשרות לחזור לתפריט הראשי.

**תפריט:** פורום, צ'אט, שיתוף קבצים, חברים עוקבים חשבון אישי.

### פורום:

1. למשתמש תהא אפשרות לפתוח אשכול
2. למשתמש תהא אפשרות להגיב לאשכול
3. למשתמש תהא אפשרות לחפש אשכול מסויים
4. כל הנ"ל למשתמש רשום משא"כ משתמש מבקר יכול לראות את נושא האשכול וזהו.

### צ'אט:

1. כאשר המשתמש יבחר בפונקציית הלייב צ'אט יפתח ממשק הדומה בצורתו לממשק ווטסאפ בשיחה קבוצתית.
2. כל ההודעות מכל המשתמשים שכרגע נמצאים בדף ירוצו על הדף.
3. המשתמש יוכל לכתוב הודעה משלו או להגיב להודעה קיימת. כמו בווטאסאפ.
4. המשתמש יוכל לסמן הודעות מסויימות כהודעות פוגעניות
5. המשתמש יוכל לחזור אחורה ולראות הודעות שקדמו באמצעות גלילה למעלה.
6. משתמש מבקר לא יוכל לסמן הודעות פוגעניות

### שיתוף קבצים:

1. אפשרות לגרור קובץ ולעלות.
2. רשימת כל הקבצים שעלו, העלאה כוללת שם המעלה ותאריך והודעה. כפתורים: העלאת קובץ, דיווח כפוגעני ודירוג הכפתורים הנ"ל, אינם פעילים למשתמש מבקר.

## **דיווח:**

בעת לחיצה על "דווח" המערת תחזיר משוב לגבי הדיווח.

## **צפייה פרופיל:**

בעת לחיצה על שם משתמש שהגיב באחד מהמקומות באתר יפתח חלון צפייה בפרופיל הכולל: מספר עוקבים, חברים, מספר פוסטים שהעלה. וכפתורים: הוסף/הסר חבר, הוסף/הסר עוקב, הכפתורים הנ"ל, אינם פעילים למשתמש מבקר.

**חברים:** רשימת החברים שלי, בעת לחיצה על אייקון של הגדרות תפתח אופציה למחיקת חבר, לא קיים אצל מבקר

**נעקבים:** כפתור נעקבים: יציג את רשימת הנעקבים שלי בעת לחיצה על אייקון של הגדרות תפתח אופציה למחיקת נעקב. כפתור עוקבים: יציג את רשימת העוקבים שלי. לא קיים אצל המבקר

## ***End point***

**משתמש:** הרשמה, התחברות

**פורום:** רשימת שרשורים, רשימת תגובות לשרשור, שרשור חדש, תגובה לשרשור.

**צ'אט:** הצגת הודעות, שליחת הודעה.

**העלאת קבצים:** הצגת העלאות, העלאת קובץ.

**דיווח:** דווח כפוגעני.

**דירוג:** דירוג הודעות.

**עוקבים/חברים:** יצירת/הסרת חבר/נעקב רשימת חברים/עוקבים/נעקבים.

**פרופיל:** פרטי משתמש.

## UX/UI- menu



# UX/UI – user

→ Login

≡ ×

name

password

Submit

→ Registration

≡ ×

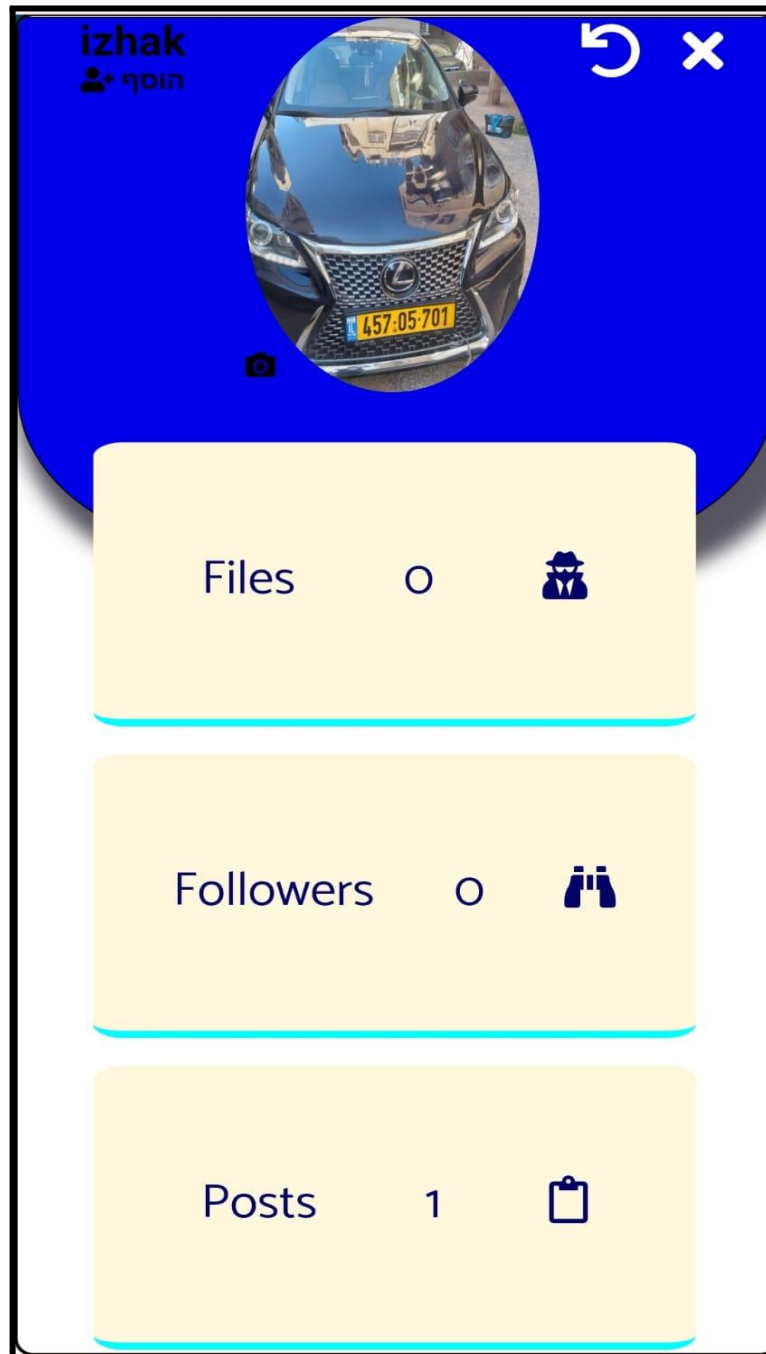
name

email

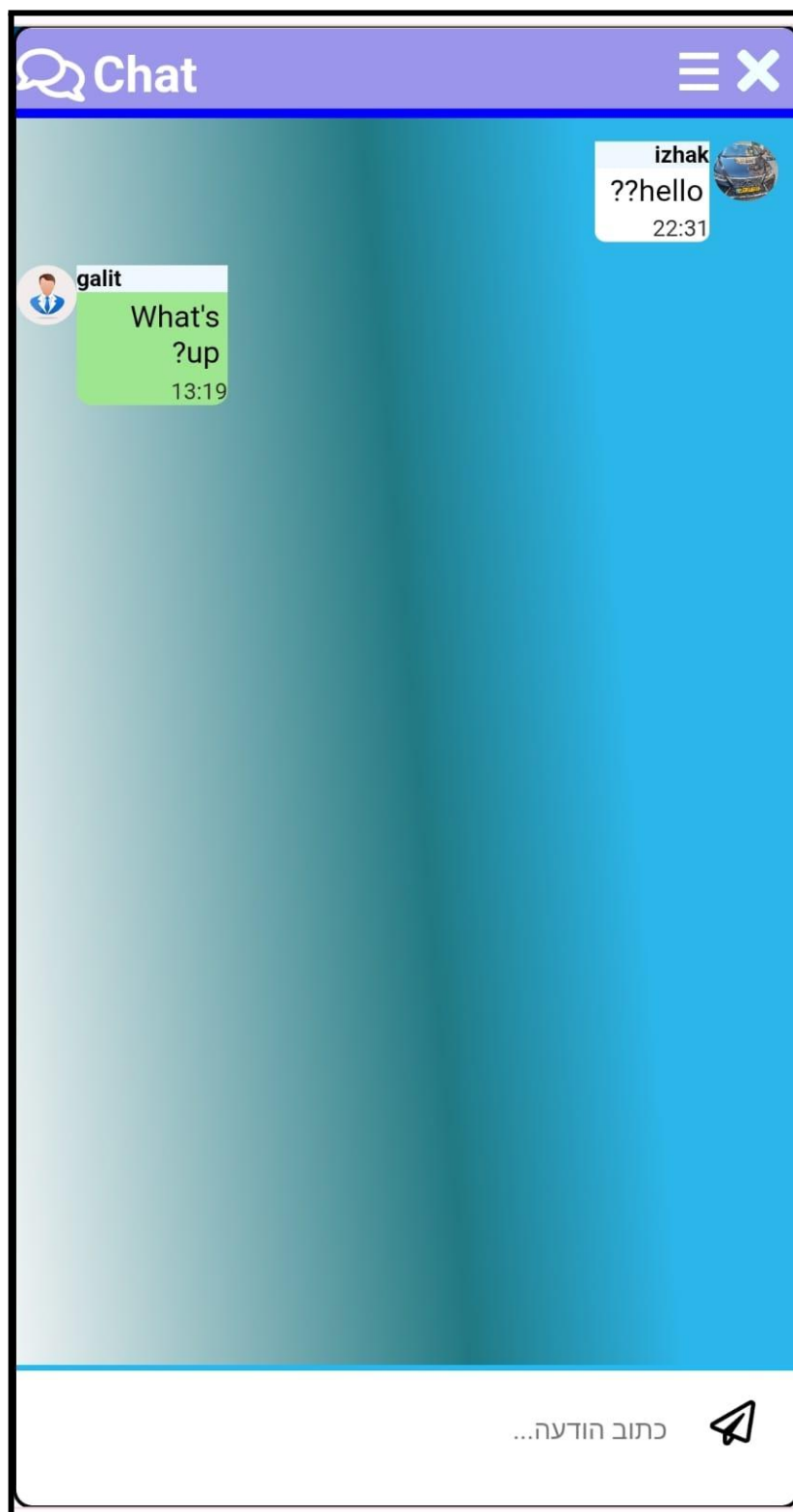
password

Submit

# UX/UI – profile

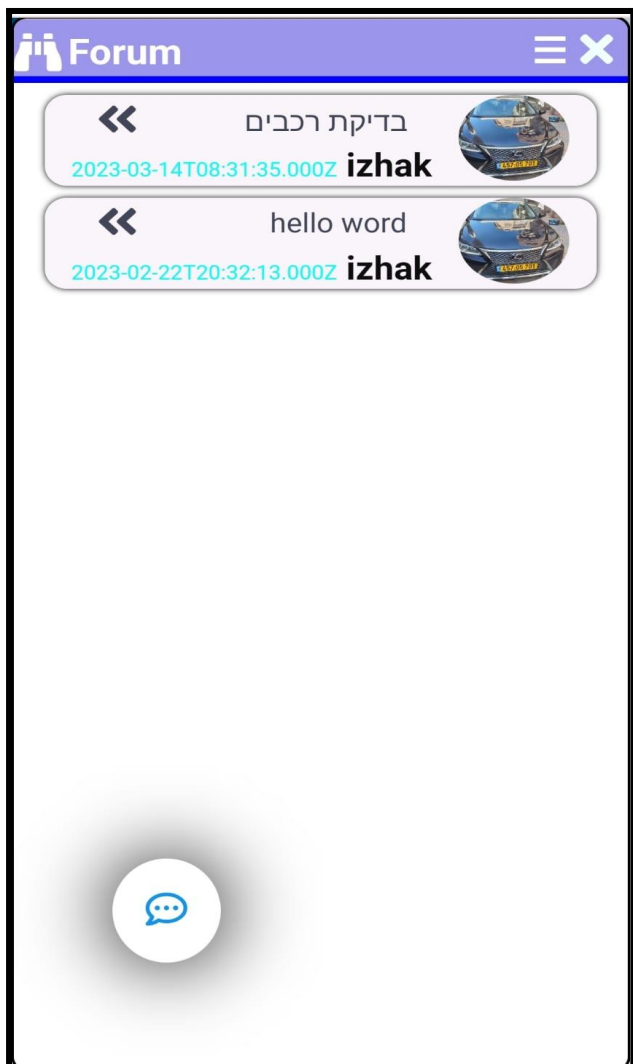


# UX/UI – chat

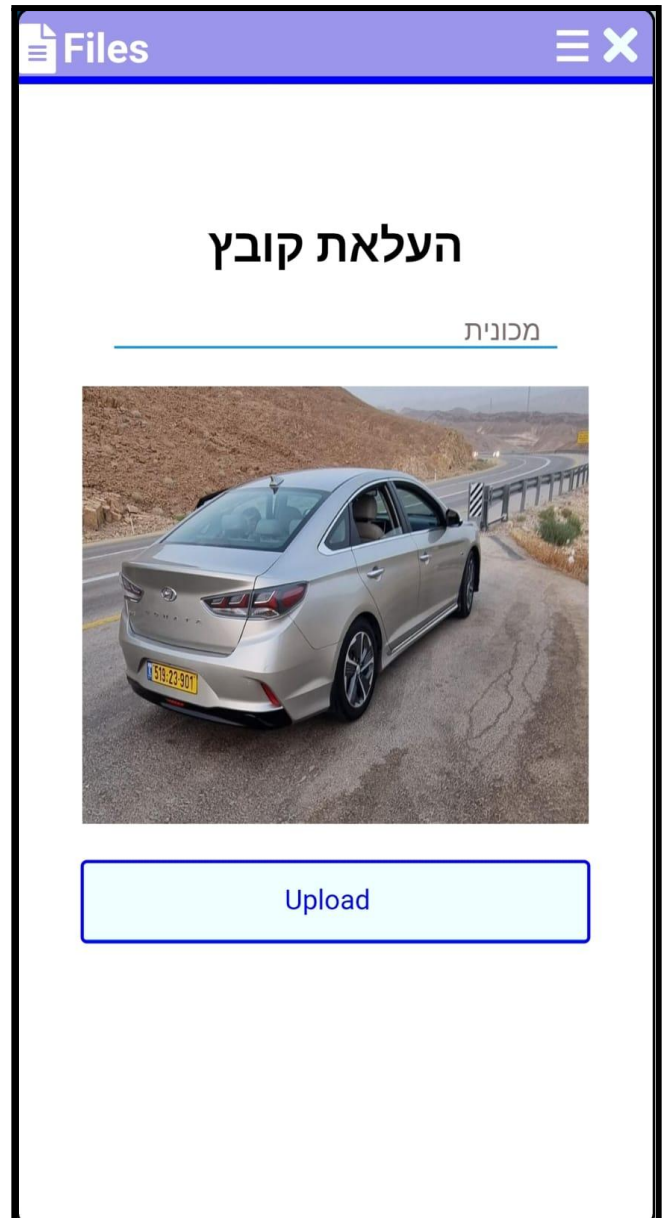
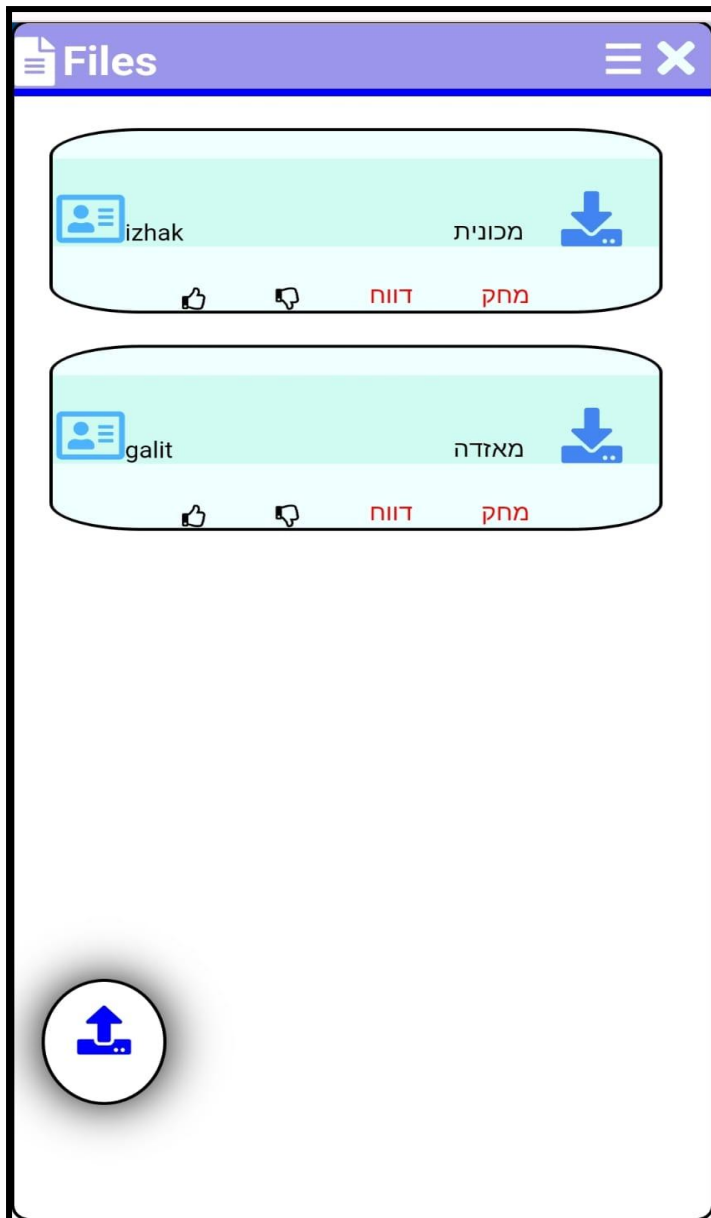




# UX/UI – forum



# UX/UI – file Sharing



# UX/UI – follow



# UX/UI – friends



## User - end point

end point	send	return
sign up	{name, email, password,}	<b>Success:</b> add user to data base, and send email to verify user <b>Failure:</b> { Error details }
validation email	{randomNumber}	Success: do active user
log in	{name or email , password}	Success: return { token } and connection with socket

## Chat- endpoint

end point	send	return
Select a correspondence window	{idUser,or null}	<b>Success:</b> Sends all correspondence,name ,image,dates <b>Failure:</b> { Error details }
send message	{idUser,message}	{id, image, message, name, userId}

## Forum - End point

End point	send	return
Thread list: get all the threads	{idUser}	<b>Success:</b> [{ id, userId, responseName, image, title, message, rating }]
New thread: create thread	{title, message, idUse}	{date, id, image, message, title}
Comment list: get all the comments	{parentId}	[{date, id, image, messages, name, rating, reply, userId}]
New comment: create comment	{id, parent}	{ date, id, image, message, name, parent, rating, replyIdForm, userId }

## File sharing- End point

end point	send	return
List uploads: get all uploads	{ token or undefined}	[[ id, user, image, userId, fileType, fileName, rate, date]]
Upload: upload file	{, fileName, fileType,}	{ date, fileName, fileType, id, image, rate, user, userId

## Offensive- End point

end point	send	return
offensive: get all the threads	{token, offensived, room}	Adds a single report number for this message, And if reported more than three times then the message will be deleted

## Rating - End point

end point	send	return
rating: Message rating	{ token, idData, room, statusRate }	Rank the message by adding or dropping a number

## Friends - End point

end point	send	return
Create friend	{ token, profileId }	Creates companies
Remove friend	{ token, profileId }	Cancels membership
friends list: get a list of friends	"{token}"	[{name,id,image}]



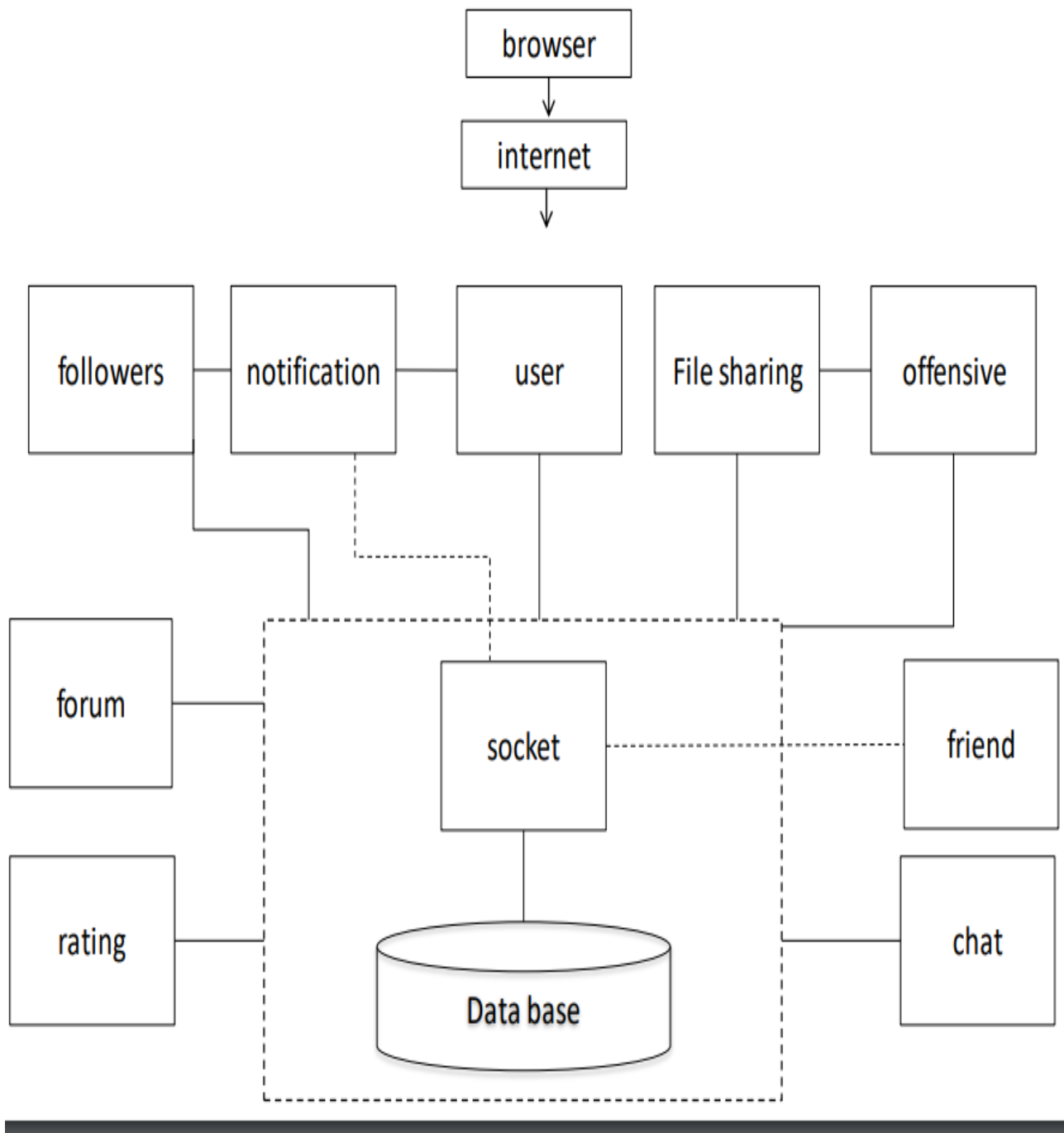
## Followers - End point

End point	Send	Return
Create followed	{ token, profileId }	Creates companies
Remove followed	{ token, profileId }	Remove followed
List followed	{token}	[{name,id,image}]

## Profile- End point

End point	Send	Return
Profile	{ token, userId }	{ followers, friends, image, name, posts, userId }

# Block diagram - server



## Class - server

### User

```
public singUpSystem( { name, password, email } ) ;  
public activeSystem( emailUser,randomNumber ) ;  
public loginSystem( { name , password } ) ;  
public CheckValidInformation( name , password,gmail )  
public proSystem ( userId);
```

### Forum

```
public fourumsSys(id,domain)  
function getMess ( domain, message, title, userId ) ;  
function createNewTheard (title,text "");  
function lastThread(domain);  
function commentList(lastComment, parent);  
function createComment(idParent,text)
```

### Chat

```
public chatSys( domain,) ;  
function newMess(mess,id)
```

### FileSharing

```
public upLoadSystem( userId, domain, fileName, fileTitle) ;  
function getFiles( domain, lastFile) ;  
public deleteFile( idfile) ;
```

### **Rating**

```
public toRate ( {room, dataId, statusRate) ;
```

### **Offensive**

```
public reporting( {room, offensivedId) ;
```

### **Friends**

```
public userAuth(idUser)  
public create( friendOne, friendTwo);  
function delete( friendOne, friendTwo);
```

### **Followers**

```
public mainFollowSystem()  
function create( friendOne, friendTwo);  
function delete( friendOne, friendTwo)
```

## Sequence diagram – Module user

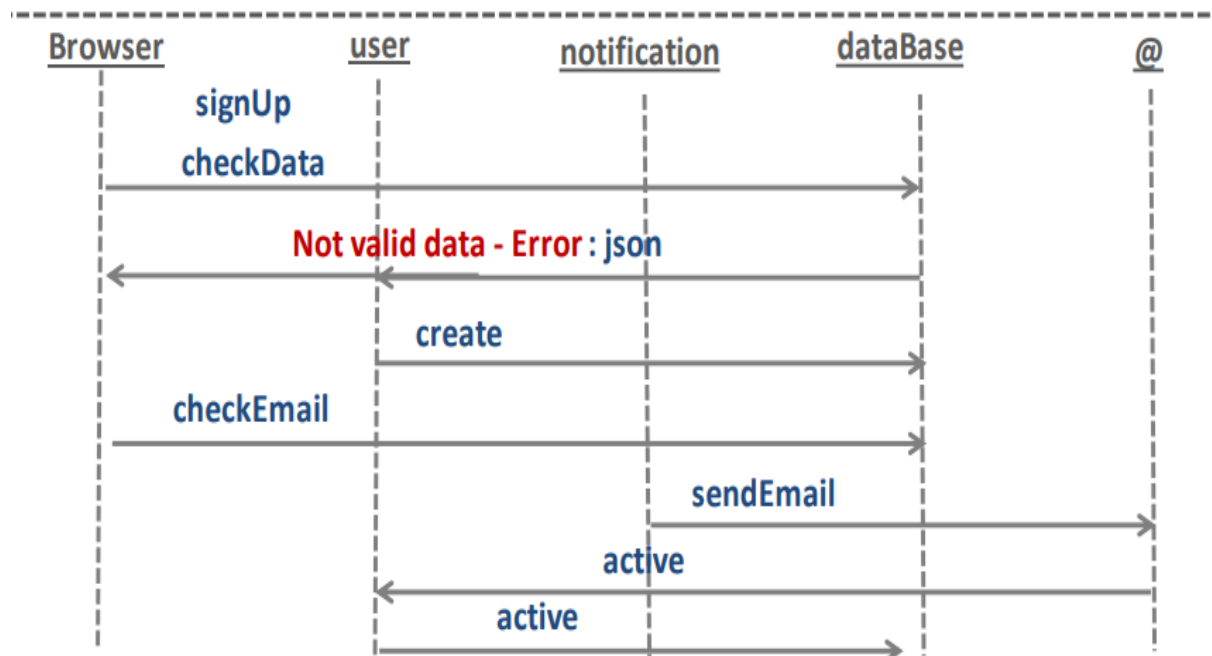
### Sign up

#### User

```
signUp = public singUpSystem ( { name, password, email } ) ;  
checkData = private checkValidInformation( { name or password or  
email } ) ;  
create = function createUser( { name, password, email } ) ;  
checkEmail = private checkEmailValid (email);  
file = public checkLegalFile();  
active = public activeSystem( emailUser,randomNumber );
```

#### Notification

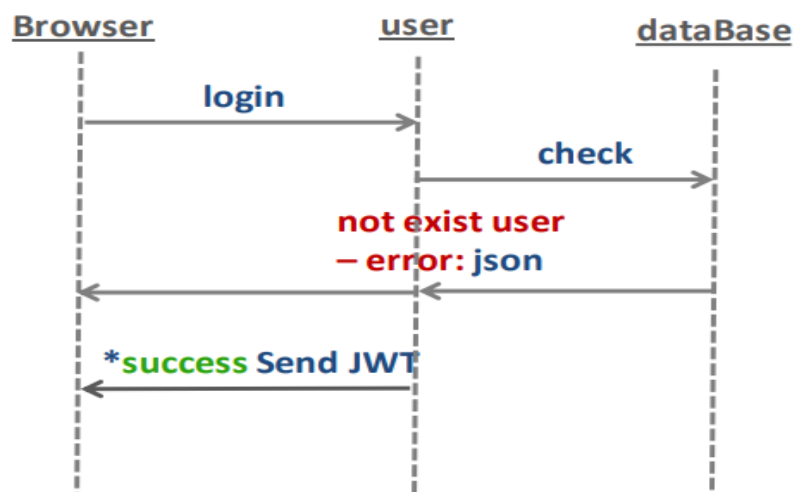
```
sendEmail = function sendEmail ( to, subject, html )
```



## Sequence diagram – Module user – log in

### User

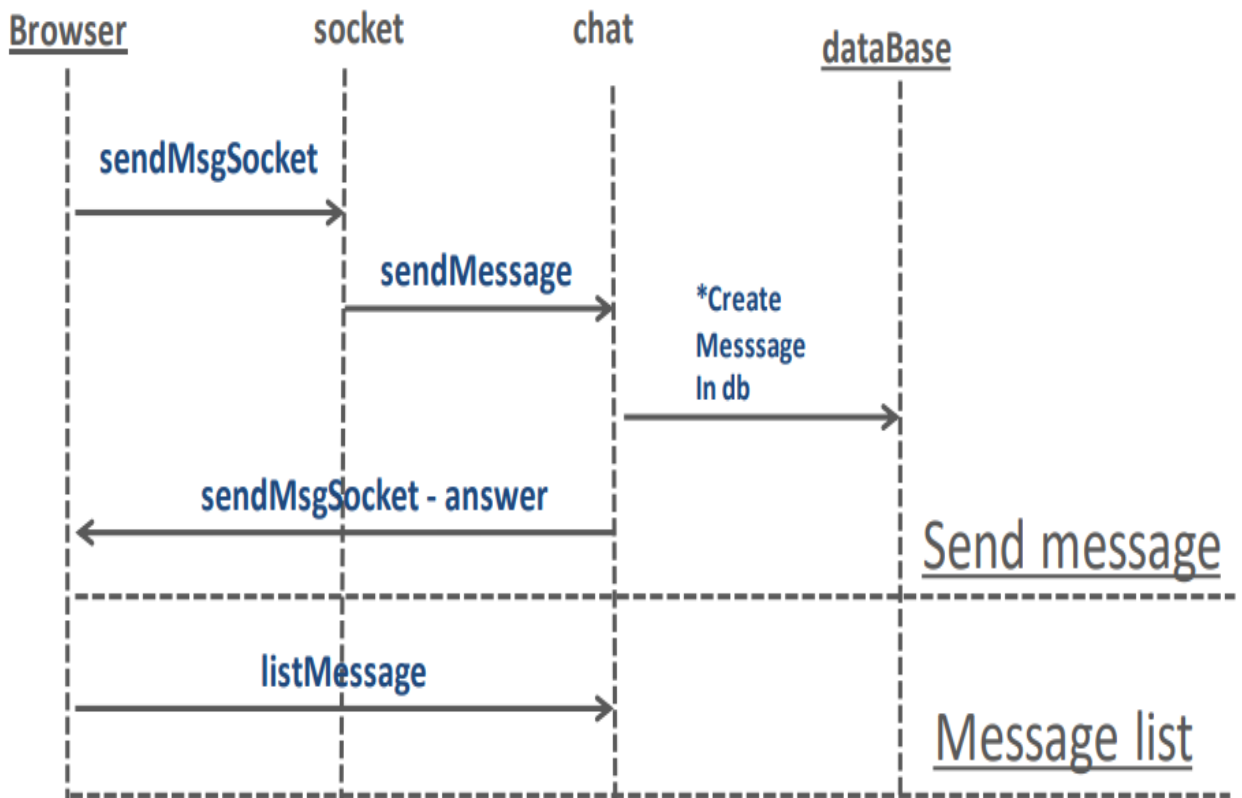
```
login = public loginSystem ( { name or email, password } );  
check = public CheckValidInformation( { name or password } );
```



## Sequence diagram – Module chat

### chat

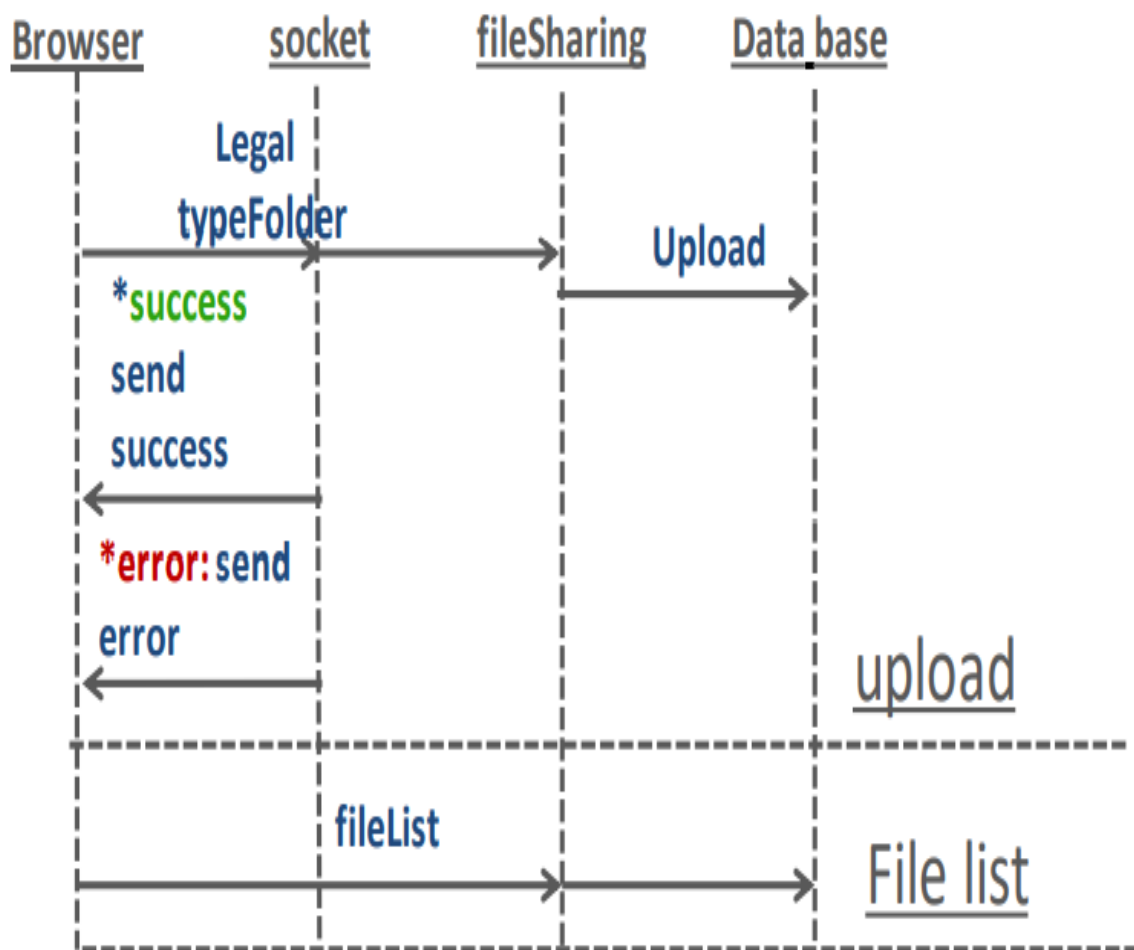
```
listMessage = public chatSys ( domain) ;  
sendMessage = function newMess( userId, message, domain)
```



## Sequence diagram – Module file sharing

### file Sharing

Upload = public uploadFile( userId, domain, fileName, fileType) ;  
fileList = function getFiles( domain) ;  
remove file = public deleteFile( idFile)

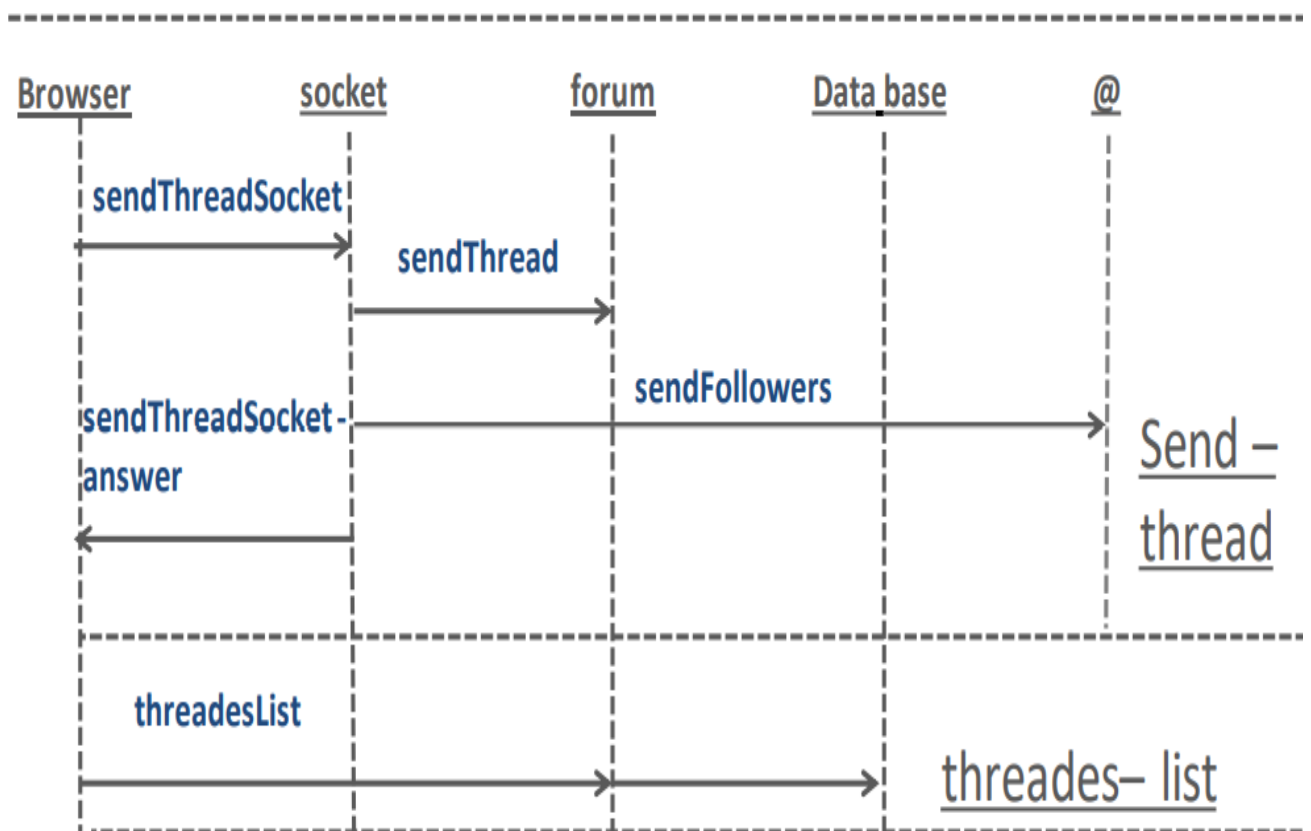




## Sequence diagram – Module

### forum (1)

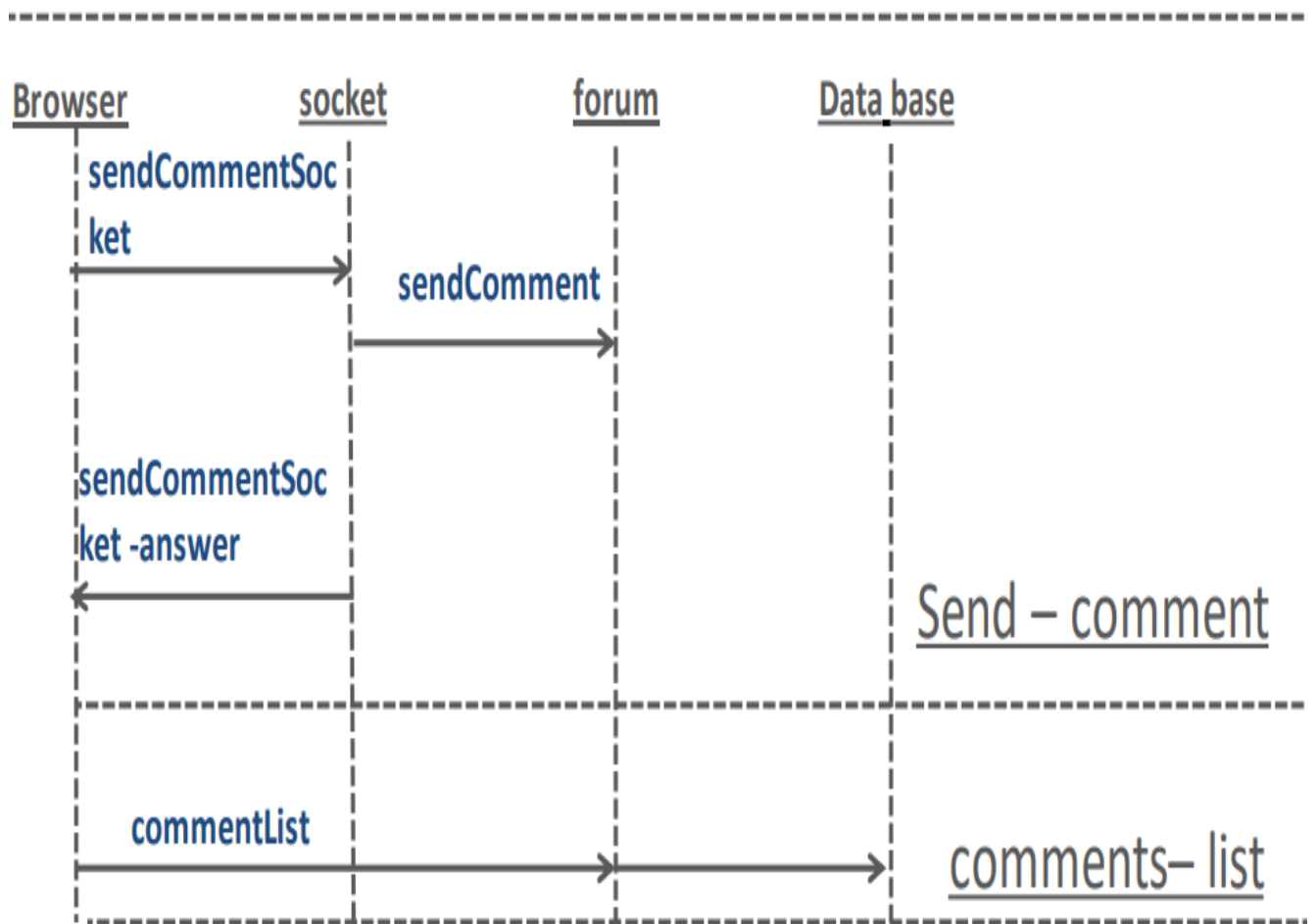
threadesList = public fourumsSys(id,domain)- function getMess ( domain, message, title, userId )  
sendThread= function createNewTheard (title,text“”);



## Sequence diagram – Module

## forum (2)

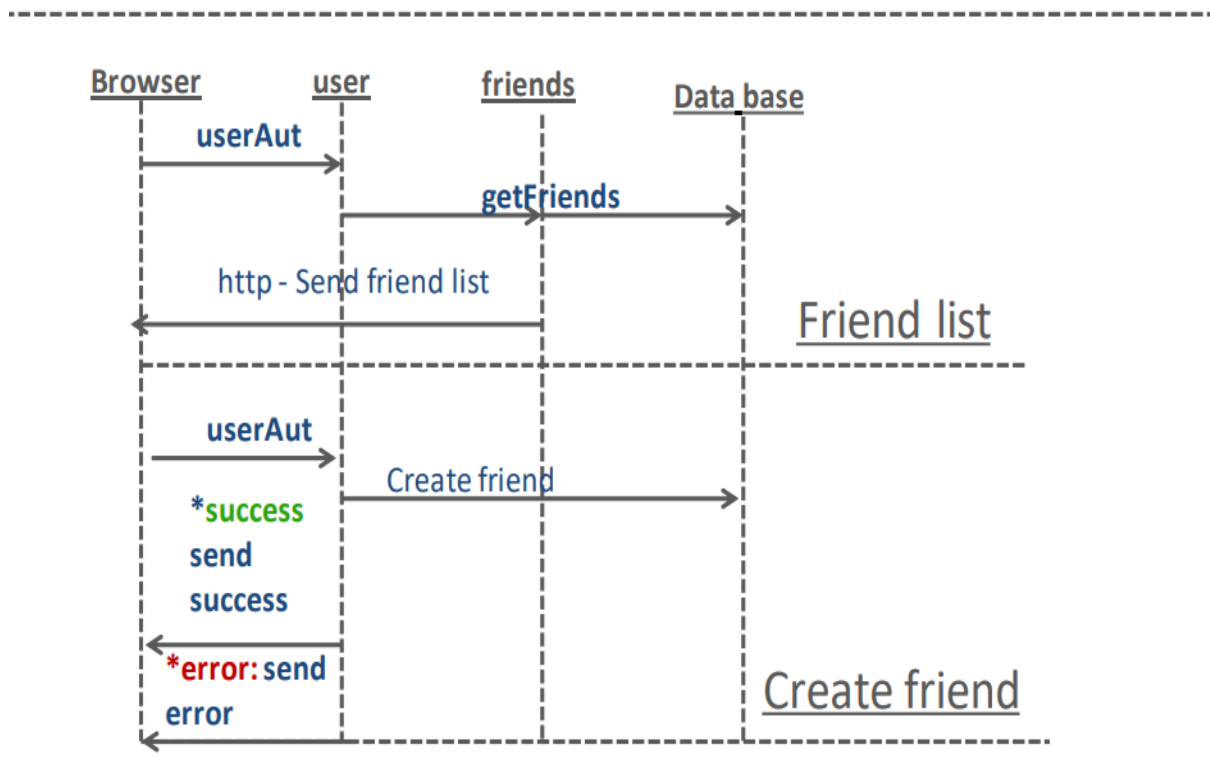
```
sendComment = function createComment(idParent,text)
commentList = function commentList(lastComment, parent);
```



## Sequence diagram – Module

### friend (1)

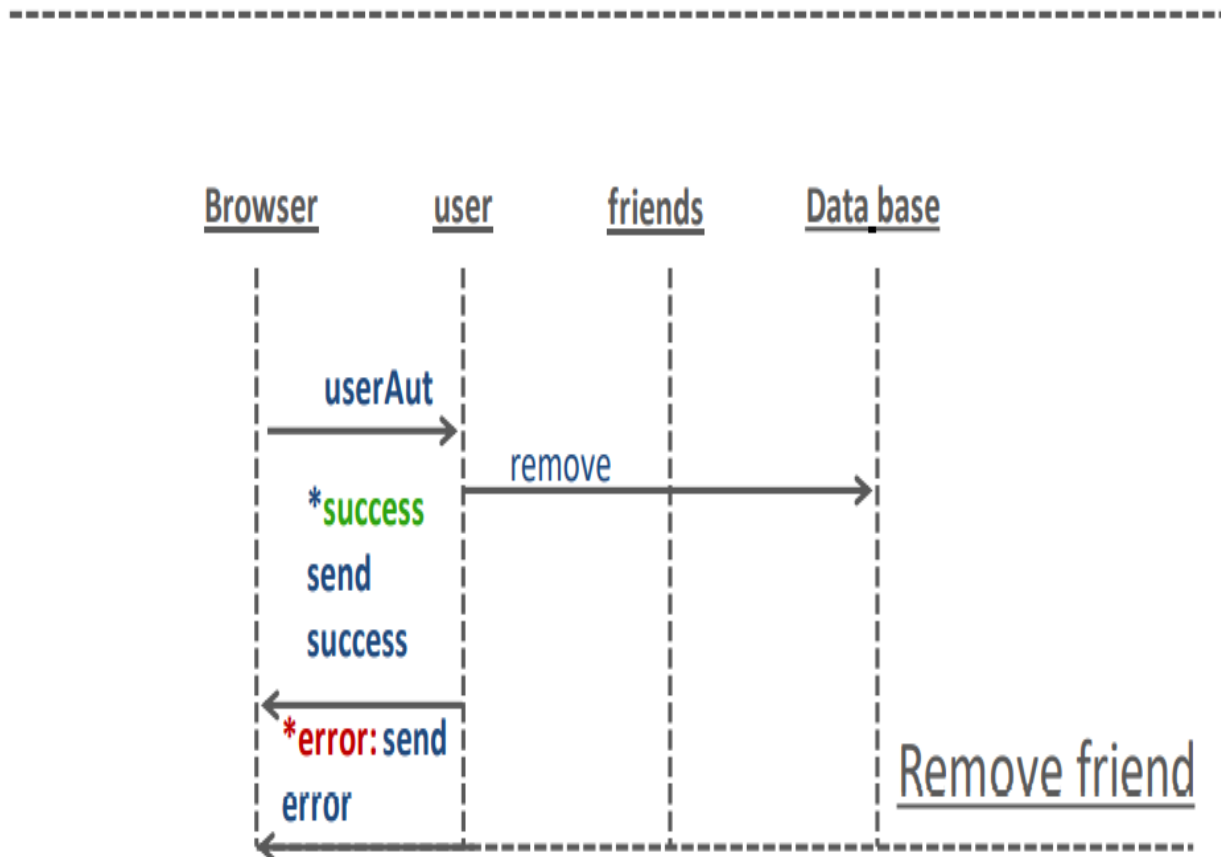
```
userAut = public userAuth( req, res, next ) ;  
getFriends = public getFriendsList ( userSession ) ;  
create = public create( friendOne, friendTwo)
```



## Sequence diagram – Module

### friend (2)

```
userAut = public userAuthenticateU ( req, res, next ) ;  
remove = public remove( friendOne, friendTwo);
```



**Sequence diagram – Module followers (1)**

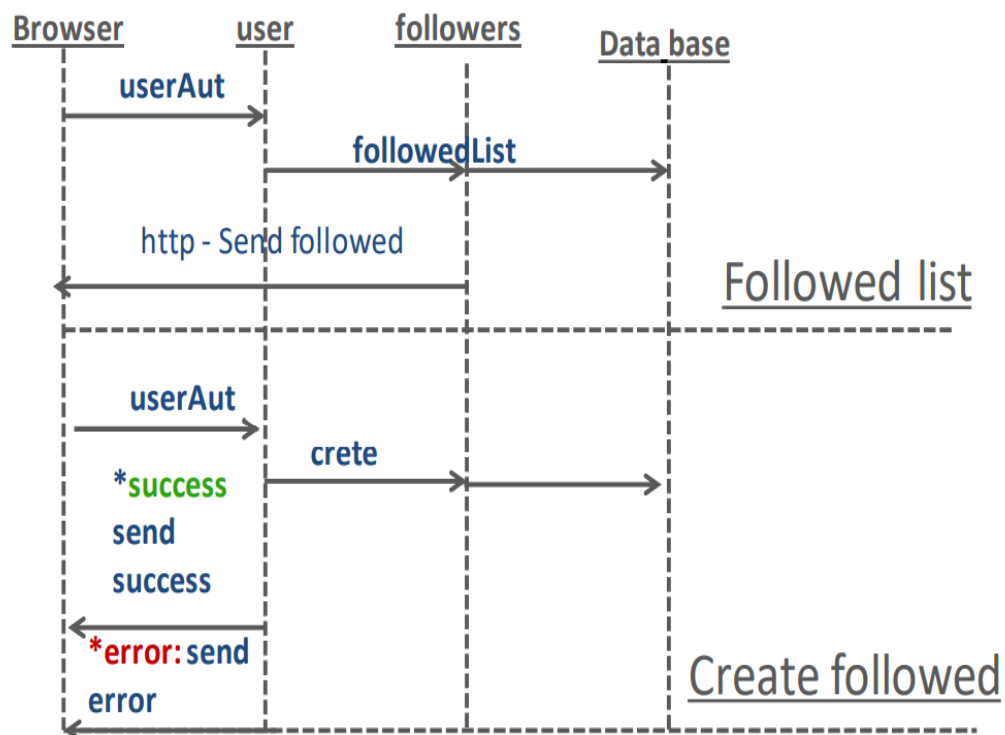
## Followers

crete = public create( followinfd, followedId) ;

followedList= public FollowedList( userId) ;

User

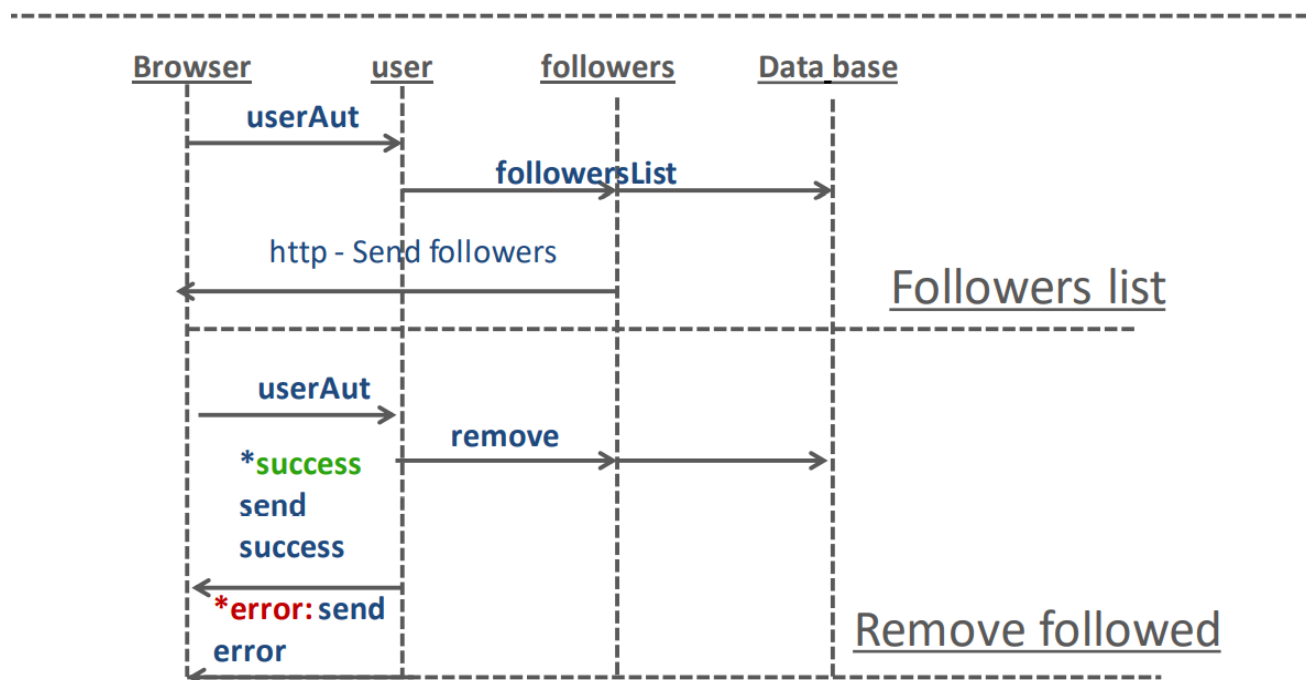
userAut = public userAuthenticateU ( req, res, next ) ;



## Sequence diagram – Module followers (2)

### Followers

```
remove= public remove( followinfld, followedId) ;  
followersList= public FollowersList( userId) ;  
User userAut = public userAuthenticateU ( req, res, next ) ;
```



## **Class - client**

### **users**

```
public loginReferences()  
public userLogin()  
public myImgORaddFriend()  
public myProfile()  
public profileReferences()  
public signReferences()  
public userSignUp()  
public checkNEP()
```

### **friends**

```
public getFriends()  
public userAut()
```

### **forum**

```
public clientFurom()  
public furomReferences()
```

### **follow**

```
public followReferences()  
public viewFollow()
```

### **files**

```
public fileReferences()  
public fileUpload()
```

### **chats**

```
public chatReferences()  
public clientChat()
```

### **additions**

```
public notifications(logo,title,texts)  
public titlesAndCookie()  
public viewThinker()
```