Περιεχόμενα

[Message 3](#_Toc9499165)

[Message( string msg, User\* pu) 3](#_Toc9499166)

[int getNumberOfLikes() 3](#_Toc9499167)

[void Like( User \*pUser) 3](#_Toc9499168)

[string toString() 3](#_Toc9499169)

[ReplyMessage 3](#_Toc9499170)

[ReplyMessage() 3](#_Toc9499171)

[User 3](#_Toc9499172)

[User( string na, string em) 3](#_Toc9499173)

[void acceptFriendRequest( bool accept) 3](#_Toc9499174)

[string getEmail() 3](#_Toc9499175)

[string getName() 3](#_Toc9499176)

[void like( string message) 3](#_Toc9499177)

[void postToWall( string message, User \*pSourceUser) 3](#_Toc9499178)

[void postToWall( string message) 4](#_Toc9499179)

[void removeFriend( User \*pUser) 4](#_Toc9499180)

[void sendFriendRequest( User \*pUser) 4](#_Toc9499181)

[string toString() 4](#_Toc9499182)

[Wall 4](#_Toc9499183)

[void postToWall( Message \*pMessage) 4](#_Toc9499184)

[string toString() 4](#_Toc9499185)

[FriendRequest 4](#_Toc9499186)

[FriendRequest( User \*pu1, User \*pu2) 4](#_Toc9499187)

[Network 4](#_Toc9499188)

[static Network \*getInstance() 4](#_Toc9499189)

[bool areFriends( User \*pUser1, User \*pUser2) 4](#_Toc9499190)

[bool areFriends( string user1, string user2) 5](#_Toc9499191)

[void clear() 5](#_Toc9499192)

[void deleteUser( User \*pUser) 5](#_Toc9499193)

[void deleteUser( string email) 5](#_Toc9499194)

[list<FriendRequest \*> \*getFriendRequestsList( User \*pUser) 5](#_Toc9499195)

[list<User \*> \*getFriendsList( User \*pUser) 5](#_Toc9499196)

[User \*getUser( string email) 5](#_Toc9499197)

[User \*insertUser( string name, string email) 5](#_Toc9499198)

[void makeFriends( User \*pUser1, User \*pUser2) 5](#_Toc9499199)

[void removeFriend( User \*pUser1, User \*pUser2) 5](#_Toc9499200)

[void sendFriendRequest( User \*pUser1, User \*pUser2) 5](#_Toc9499201)

[string toString() { 5](#_Toc9499202)

[bool userExists( string email) 5](#_Toc9499203)

[Exceptions 6](#_Toc9499204)

[NetworkException 6](#_Toc9499205)

[virtual const char \* what () 6](#_Toc9499206)

[UserExistsException 6](#_Toc9499207)

[UserExistsException( string e) 6](#_Toc9499208)

[virtual const char \* what () 6](#_Toc9499209)

[AlreadyFriendsException 6](#_Toc9499210)

[AlreadyFriendsException (User \*pu1, User \*pu2) 6](#_Toc9499211)

[virtual const char \* what () 6](#_Toc9499212)

[AlreadyLikeException 6](#_Toc9499213)

[AlreadyLikeException(User \*pu, Message \*pm) 6](#_Toc9499214)

[virtual const char \* what () 6](#_Toc9499215)

[NoFriendsException 7](#_Toc9499216)

[NoFriendsException (User \*pu1, User \*pu2) 7](#_Toc9499217)

[virtual const char \* what () 7](#_Toc9499218)

[UserNotExistsException 7](#_Toc9499219)

[UserNotExistsException( string e) 7](#_Toc9499220)

[virtual const char \* what () 7](#_Toc9499221)

[SameUserException 7](#_Toc9499222)

[SameUserException( string e) 7](#_Toc9499223)

[virtual const char \* what () 7](#_Toc9499224)

# Message

## Message( string msg, User\* pu)

Ο constructor που δέχεται σαν παραμέτρους το μήνυμα σαν string και το χρήστη ο οποίος το στέλνει.

## int getNumberOfLikes()

Επιστρέφει τον αριθμό των Likes που έχει δεχτεί το συγκεκριμένο μήνυμα.

## void Like( User \*pUser)

Ο χρήστης pUser κάνει Like στο μήνυμα

## string toString()

Επιστρέφει τα περιεχόμενα σαν string ώστε να εμφανιστούν στην οθόνη σε αναγνώσιμη μορφή.

# ReplyMessage

## ReplyMessage()

Ο constructor της κλάσης.

# User

## User( string na, string em)

Αρχικοποιεί το χρήστη δίνοντας το όνομα και το email του.

## void acceptFriendRequest( bool accept)

//TO\_DO

## string getEmail()

Επιστρέφει το email του χρήστη.

## string getName()

Επιστρέφει το όνομα του χρήστη

## void like( string message)

Κάνει like σε ένα μήνυμα

## void postToWall( string message, User \*pSourceUser)

Στέλνει στον τοίχο το μήνυμα από το χρήστη pSourceUser που μπορεί να είναι είτε ο εαυτός του είτε κάποιος φίλος του.

## void postToWall( string message)

Στέλνει στον τοίχο το μήνυμα από τον ίδιο.

## void removeFriend( User \*pUser)

Αφαιρεί από φίλο κάποιο φίλο.

## void sendFriendRequest( User \*pUser)

Δέχεται αίτημα φιλίας από το χρήστη pUser

## string toString()

Επιστρέφει τα περιεχόμενα του χρήστη με μορφή string ώστε να τυπωθούν στην οθόνη.

# Wall

## void postToWall( Message \*pMessage)

Κάνει post ένα μήνυμα

## string toString()

Επιστρέφει τα περιεχόμενα του τοίχου με μορφή string ώστε να τυπωθούν στην οθόνη.

# FriendRequest

## FriendRequest( User \*pu1, User \*pu2)

Ο constructor που δέχεται σαν παραμέτρους δύο χρήστες από τους οποίους ο πρώτος κάνει αίτημα φιλίας στο δεύτερο.

# Network

## static Network \*getInstance()

Στην πρώτη φορά που καλείται δημιουργεί ένα αντικείμενο της κλάσης Network και επιστρέφει pointer σε αυτό. Στις επόμενες κλήσεις επιστρέφει τον ίδιο pointer.

## bool areFriends( User \*pUser1, User \*pUser2)

Επιστρέφει true ή false ανάλογα με το αν δύο χρήστες pUser1 και pUser2 είναι φίλοι.

## bool areFriends( string user1, string user2)

Επιστρέφει true ή false ανάλογα με το αν δύο email αντιστοιχούν σε χρήστες που είναι φίλοι.

## void clear()

Διαγράφει όλα τα περιεχόμενα του δικτύου.

## void deleteUser( User \*pUser)

Διαγράφει ένα χρήστη δίνοντας τον pointer σε αυτό το χρήστη.

## void deleteUser( string email)

Διαγράφει ένα χρήστη δίνοντας το email του.

## list<FriendRequest \*> \*getFriendRequestsList( User \*pUser)

Επιστρέφει τη λίστα με τα αιτήματα φιλίας προς ένα χρήστη pUser

## list<User \*> \*getFriendsList( User \*pUser)

Επιστρέφει τη λίστα των φίλων ενός χρήστη pUser

## User \*getUser( string email)

Δίνοντας το email επιστρέφει το χρήστη στον οποίο αντιστοιχεί αυτό το email.

## User \*insertUser( string name, string email)

Προσθέτει ένα καινούριο χρήστη στο δίκτυο παίρνοντας σαν παραμέτρους το όνομα και το email του.

## void makeFriends( User \*pUser1, User \*pUser2)

Κάνει φίλους τους χρήστες pUser1 και pUser2

## void removeFriend( User \*pUser1, User \*pUser2)

Αφαιρεί από φίλους τους χρήστες pUser1 και pUser2

## void sendFriendRequest( User \*pUser1, User \*pUser2)

Στέλνει αίτημα φιλίας ο χρήστης pUser1 στο χρήστη pUser2

## string toString() {

Επιστρέφει με μορφή string τα περιεχόμενα του δικτύου.

## bool userExists( string email)

Επιστρέφει true αν το email αντιστοιχεί σε χρήστη που υπάρχει.

## 

# Exceptions

## NetworkException

Κλάση για τα exceptions. Είναι η βασική κλάση από την οποία κληρονομούνται όλες οι άλλες κλάσεις exceptions του project.

### virtual const char \* what ()

Επιστρέφει με μορφή string τα στοιχεία του exception.

### UserExistsException

Exception για την περίπτωση που γίνεται προσπάθεια εισαγωγής στη βάση χρήστη που υπάρχει ήδη.

### UserExistsException( string e)

Η πρώτη παράμετρος του exception είναι το email του χρήστη.

### virtual const char \* what ()

Επιστρέφει με μορφή string τα στοιχεία του exception.

## AlreadyFriendsException

Exception για την περίπτωση που γίνει προσπάθεια να γίνει φίλος κάποιος που είναι ήδη φίλος.

### AlreadyFriendsException (User \*pu1, User \*pu2)

Η πρώτη παράμετρος του exception είναι ο χρήστης pu1 που προσπαθεί να κάνει φίλο τον χρήστη pu2.

### virtual const char \* what ()

Επιστρέφει με μορφή string τα στοιχεία του exception.

## AlreadyLikeException

Exception για την περίπτωση που γίνει προσπάθεια για Like σε μήνυμα που έχει ήδη γίνει Like.

### AlreadyLikeException(User \*pu, Message \*pm)

Η πρώτη παράμετρος του exception είναι ο χρήστης και η δεύτερη το μήνυμα

### virtual const char \* what ()

Επιστρέφει με μορφή string τα στοιχεία του exception.

## NoFriendsException

Exception για την περίπτωση που γίνει προσπάθεια ενέργειας που προορίζεται μόνο για φίλους (π.χ. αποστολή μηνύματος) σε κάποιον που δεν είναι φίλος.

### NoFriendsException (User \*pu1, User \*pu2)

Η πρώτη παράμετρος του exception είναι ο χρήστης pu1 που προσπαθεί να κάνει ενέργεια στον χρήστη pu2.

### virtual const char \* what ()

Επιστρέφει με μορφή string τα στοιχεία του exception.

## UserNotExistsException

Exception για την περίπτωση που γίνεται προσπάθεια πρόσβασης σε χρήστη (χρησιμοποιώντας το email του) που δεν υπάρχει.

### UserNotExistsException( string e)

Η πρώτη παράμετρος του exception είναι το email του χρήστη.

### virtual const char \* what ()

Επιστρέφει με μορφή string τα στοιχεία του exception.

### SameUserException

Exception για την περίπτωση που γίνει προσπάθεια να κάνει κάποιος φίλο τον εαυτό του.

### SameUserException( string e)

Η πρώτη παράμετρος του exception είναι ο χρήστης.

### virtual const char \* what ()

Επιστρέφει με μορφή string τα στοιχεία του exception.