

Progetto di Basi di Dati

THE TEK FORUM

Cusinato Giacomo - 1054137

Callegaro Michael - 1051789

A.A. 2013/2014

1 Abstract

Il progetto consiste nella creazione di una base di dati che consenta la gestione completa di un semplice forum incentrato sulle nuove tecnologie hardware riguardanti i personal computer. Come ogni community virtuale, il forum andrà ad ospitare numerosi utenti che potranno consultare tutte le sezioni e discussioni aperte nel forum e, previa registrazione, interagire con la piattaforma aggiungendo nuovi thread, intervenendo in discussioni già presenti o scambiandosi messaggi privati. Sono inoltre presenti più figure tra gli utenti: i moderatori con i privilegi di modificare o cancellare qualsiasi thread o post e gli admin che, oltre ad avere i poteri dei mod, potranno gestire il forum modificando o rimuovendo sia le varie sezioni che tutti gli utenti registrati.

2 Analisi dei requisiti

Il progetto mira alla realizzazione di un database che contenga e gestisca le risorse di un forum e che elabori le operazioni richieste dall'utente tramite l'interfaccia dell'applicazione web che fa uso della base di dati stessa. Il forum è principalmente composto da: utenti, sezioni, thread, post e discussioni secondo delle specifiche associazioni.

Gli utenti vengono identificati da uno username che è unico per tutto il forum e varrà come identificativo per qualsiasi intervento nel forum. Vengono inoltre richieste una mail ed una password per la registrazione, un contatore dei post (che verrà incrementato a seconda dell'attività dell'utente) ed una data di registrazione. Esistono anche due tipi di utenti specifici: i moderatori sono di fatto utenti che possono modificare post o thread, mentre gli admin hanno anche i privilegi per gestire sezioni ed utenti.

Le sezioni sono essenzialmente dei contenitori globali di discussioni accumulate da un argomento specifico. Sono identificate da un titolo e possono contenere una descrizione opzionale. Ogni sezione, inoltre, può contenere più sottosezioni con le stesse proprietà della sezione madre.

I thread sono discussioni iniziate da un utente registrato in una particolare sezione. Sono caratterizzate da un titolo, un contenuto opzionale ad una data di creazione. Un thread è correlato esclusivamente ad un utente ed una sezione specifica. Vengono forniti inoltre dettagli per ogni discussione aperta, quali numero di post ed ultime eventuali modifiche da parte di admin o moderatori.

I post sono dei messaggi che vengono inseriti come risposta ai thread aperti dai vari utenti registrati al forum. Ogni post è caratterizzato da un contenuto ed una data di creazione e collegato ad un specifico utente ed un particolare thread. Ogni nuovo post aumenta il contatore messaggi dell'autore.

Le conversazioni sono contenitori di messaggi privati scambiati tra i vari utenti registrati. Una conversazione può essere creata da un utente ma più persone possono parteciparvi. Sono quindi caratterizzate da un oggetto e legati a più destinatari, oltre che al mittente.

I messaggi contengono il testo vero e proprio scritto dai partecipanti in una determinata conversazione. Sono caratterizzati da un contenuto (il testo del messaggio), una data di creazione e legati ad un utente ed una conversazione.

3 Progettazione concettuale

3.1 Lista delle classi

Users

- Usernamestring <<PK>>: nome univoco di un utente, chiave primaria
- Pwdstring: password dell'utente per l'accesso al forum
- RegDatedate: string di registrazione dell'utente.
- PostCountint: numero di post effettuati dall'utente nel forum

Moderators

- Usernamestring <<PK>>: nome univoco di un moderatore, chiave primaria

Admins

- Usernamestring <<PK>>: nome univoco di un admin, chiave primaria

Sections

- Namestring <<PK>>: nome univoco di una sezione, chiave primaria
- Descriptionstring: descrizione delle sezione
- SupSectionstring: se sottosezione, indica il nome della sezione madre

Threads

- Idint <<PK>>: id univoco di una thread, chiave primaria
- Titlestring: titolo del thread
- Contentstring: contenuto del thread
- PostDatedate: data di creazione del thread

Posts

- Idint <<PK>>: id univoco di una post, chiave primaria
- Contentstring: contenuto del post
- PostDatedate: data di creazione del post
- Usernamestring: utente autore del post

Conversations

- Idint <<PK>>: identificatore univoco di una conversazione privata, chiave primaria
- Objectstring: oggetto della conversazione

Messages

- Idint <<PK>>: identificatore univoco di un messaggio, chiave primaria
- Contentstring: testo del messaggio
- PostDatedate: data di invio del messaggio

ThreadDetails

- Idint <<PK>>: identificatore univoco, chiave primaria
- LastModified date: ultima data registrata della modifica modifcaead
- LastMod string: autore dell'ultima modifica

3.2 Descrizione delle Associazioni

Users -Threads ha scritto

- Un utente può pubblicare più thread. Un thread è pubblicato da un utente.
- Molteplicità: 1 - N
- Totalità: parziale verso Threads e totale verso Users

ThreadDetails -Threads è caratterizzato da

- Un thread ha una specifica di dettagli, una specifica di dettagli appartengono ad un thread
- Molteplicità: 1 - 1
- Totalità: totale verso entrambi

Sections -Threads contiene

- Una sezione può contenere più thread, un thread si trova in una sola sezione
- Molteplicità: 1 - N
- Totalità: parziale verso Threads, totale verso Sections

Threads -Posts contiene

- Una thread può contenere più post, un post si trova in un solo thread
- Molteplicità: 1 - N
- Totalità: parziale verso Posts, totale verso Threads

Sections - Sections è sottosezione di

- Una sezione può avere più di una sottosezione
- Molteplicità: 1 - N
- Totalità: parziale verso Sections (sottosezione), totale verso Threads

Users - Conversations partecipa

- Un utente può partecipare a più conversazioni, una conversazione contiene più utenti
- Molteplicità: N -M
- Totalità: parziale verso conversations e totale verso Users

- Nuova tabella: UserConversation

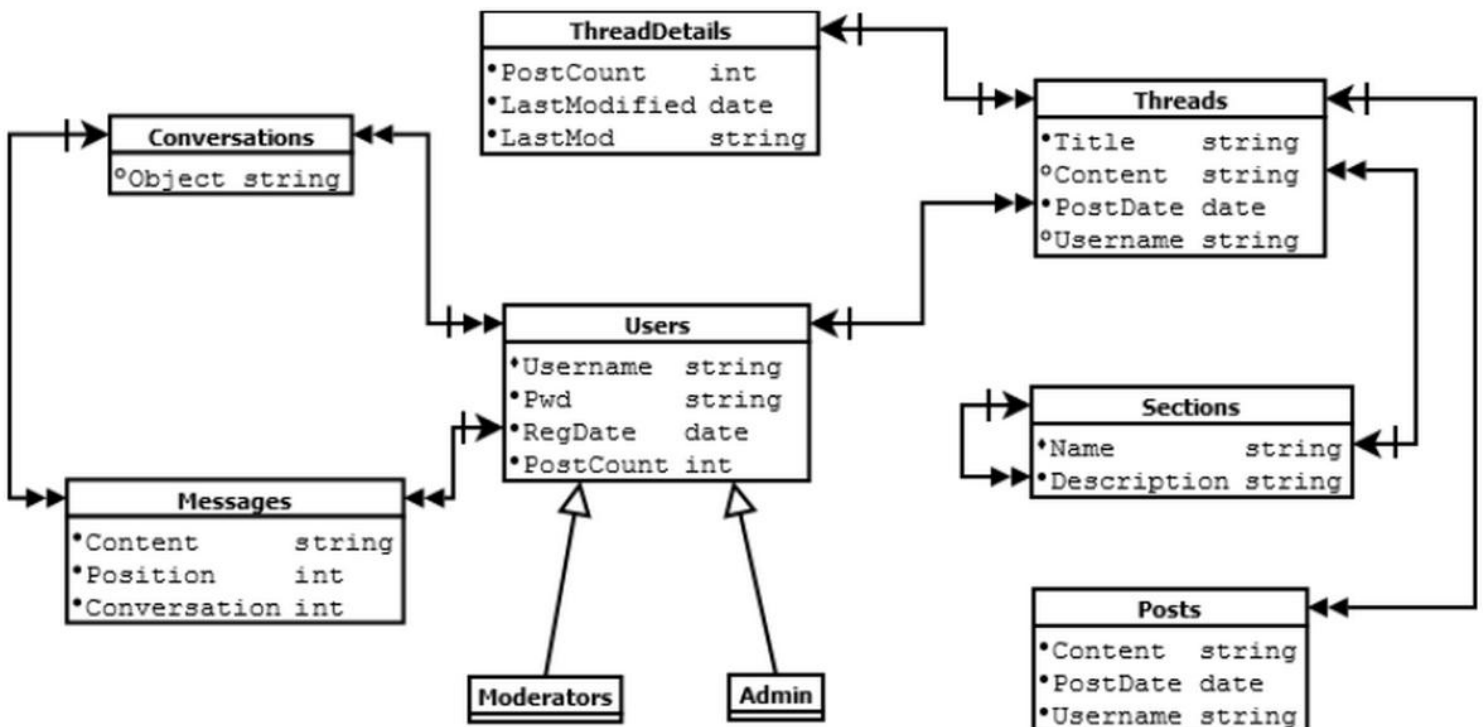
Conversations - Messages contiene

- Una conversazione contiene più messaggi, un messaggio si trova in una conversazione
- Molteplicità 1 - N
- Totalità: totale verso Conversations, parziale verso Messages

Users - Messages ha scritto

- Un utente può scrivere più messaggi, un messaggio è scritto da un utente
- Molteplicità 1 - N
- Totalità: totale verso Users, parziale verso Messages

3.3 Rappresentazione grafica dello schema concettuale



4 Progettazione Logica

4.1 Descrizione testuale dello schema relazionale

Users (Username: string, Pwd: sting, Email: string, RegDate: date, PostCount: int)

- PK: Username
- DEFAULT PostCount = 0

Admins (Username: string)

- FK: Username - Username(Users)

Moderators (Username: string)

- FK Username - Username(Users)

Threads (Id: int, Title: string, Content: string, PostDate: date, Username: string, SectionName: string)

- PK: Id
- NULLABLE Content
- FK: Username - Username(Users)
- FK: SectionName - Name(Sections)

ThreadDetails (Id: int, PostCount: int, LastModified: date, LastMod: string)

- PK: Id
- FK: Id - Id(Thread)
- DEFAULT PostCount = 0
- NULLABLE LastModified, LastMod

Sections (Name: string, Description: string, SupSection: string)

- PK: Name
- NULLABLE Description

- FK: SupSection - Name(Sections)

Posts (Id: int, Content: string, PostDate: date, Username: string, ThreadId: int)

- PK: Id
- FK: Username - Username(Users)
- FK: ThreadId - Id(Threads)

Conversations (Id: int, Object: string)

- PK: Id

UserConversation (Username: string, Id: int)

- FK: Username - Username(Users)
- FK: Id - Id (Conversations)

Messages (Id: int, Content: string, PostDate: date, Username: string, Conversation: int)

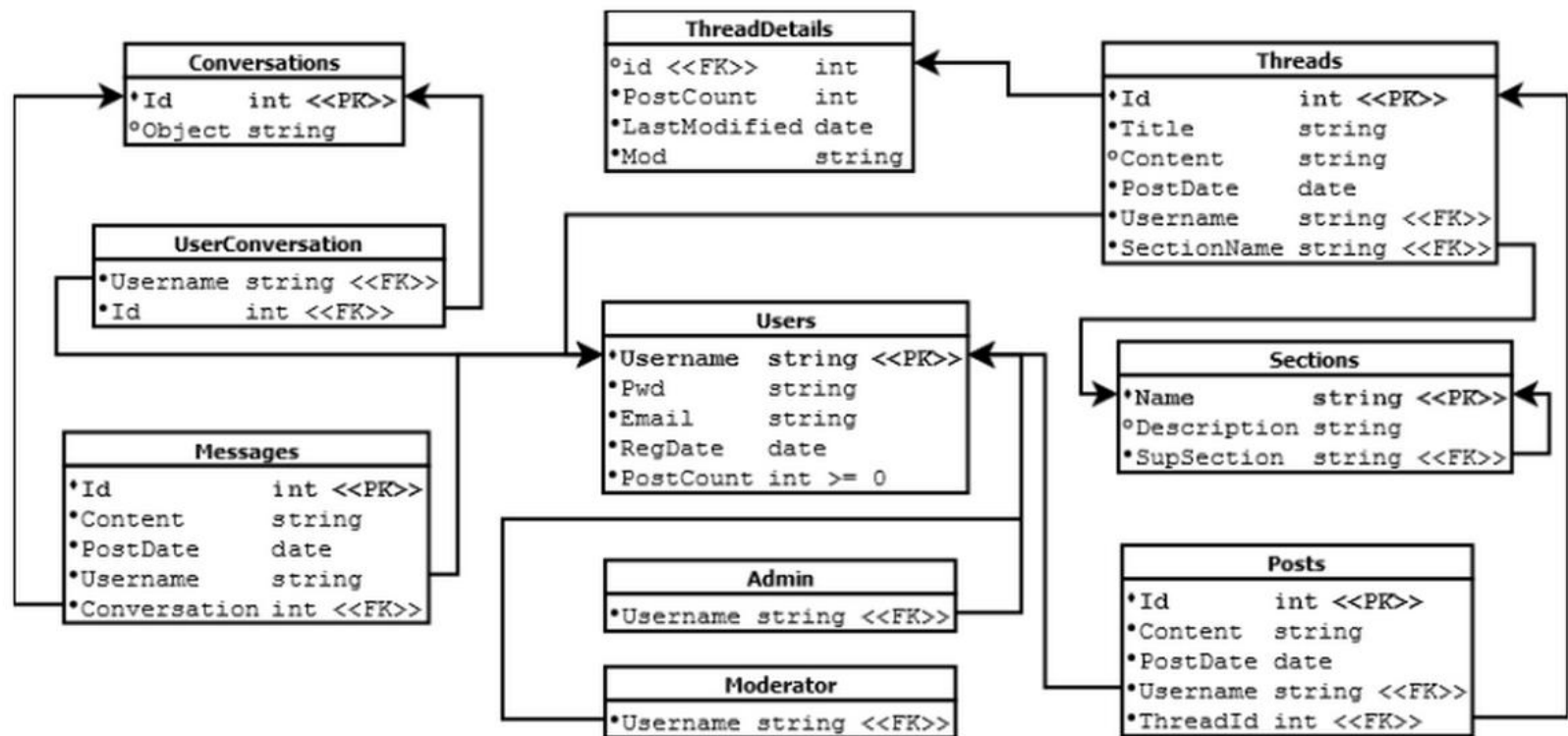
- PK: Id
- FK: Username - Username(Users)
- FK: Conversation - Id (Conversations)

4.2 Considerazioni sulle scelte progettuali

Nel modello logico è stato scelto di rappresentare la gerarchia di utenti tramite partizionamento verticale. È stata quindi mantenuta la superclasse Users ed il vincolo referenziale dalle due nuove relazioni (Admins e Moderators) che contengono esclusivamente l'identificatore di un particolare utente.

L'associazione molti a molti tra Conversations e Users è stata risolta aggiungendo una nuova relazione UserConversation allo schema che associa ogni conversazione ad uno o più utenti che vi partecipano. La nuova relazione contiene due chiavi esterne che riferiscono alle due relazioni coinvolte.

4.3 Rappresentazione grafica dello schema logico



5 Implementazione dello schema logico

```
/* Clean the database first */
```

```
SET FOREIGN_KEY_CHECKS=0;
```

```
DROPTABLEIFEXISTS Users;  
DROPTABLEIFEXISTS Moderators;  
DROPTABLEIFEXISTS Admins;  
DROPTABLEIFEXISTS Threads;  
DROPTABLEIFEXISTS ThreadDetails;  
DROPTABLEIFEXISTS Sections;  
DROPTABLEIFEXISTS Posts;  
DROPTABLEIFEXISTS Conversations;  
DROPTABLEIFEXISTS UserConversation;  
DROPTABLEIFEXISTS Messages;
```



```
/* Create the tables */
```

```
SET FOREIGN_KEY_CHECKS=1;
```

```
CREATETABLE Users (  
    Username VARCHAR(20) PRIMARYKEY,  
    Pwd VARCHAR(40) NOTNULL,  
    Email VARCHAR(40) NOTNULL,  
    RegDate DATENOTNULL,  
    PostCount INTNOTNULLDEFAULT0  
) ENGINE = InnoDB;
```

```
CREATETABLE Moderators (  
    Username VARCHAR(20) NOTNULL,  
    FOREIGNKEY (Username) REFERENCES Users(Username)  
        ONDELETECASCADE  
        ONUPDATECASCADE  
) ENGINE = InnoDB ;
```

```
CREATETABLE Admins (  
    Username VARCHAR(20),  
    FOREIGNKEY (Username) REFERENCES Users(Username)  
        ONDELETECASCADE  
        ONUPDATECASCADE  
) ENGINE = InnoDB;
```

```
CREATETABLE Sections (  
    Name VARCHAR(50) PRIMARYKEY,  
    Description VARCHAR(150),  
    SupSection VARCHAR(50) NOTNULLDEFAULT"root",  
    FOREIGNKEY (SupSection) REFERENCES Sections(Name)  
        ONDELETECASCADE  
        ONUPDATECASCADE  
) ENGINE = InnoDB;
```

```
CREATETABLE Threads (  
    Id INTPRIMARYKEY,  
    Title VARCHAR(100) NOTNULL,  
    Content VARCHAR(65000),  
    PostDate DATENOTNULL,  
    Username VARCHAR(20) NOTNULL,  
    SectionName VARCHAR(50) NOTNULL,  
    FOREIGNKEY (Username) REFERENCES Users(Username)  
        ONDELETECASCADE  
        ONUPDATECASCADE,
```

```

        FOREIGNKEY (SectionName) REFERENCES Sections(Name)
        ONDELETECASCADE
        ONUPDATECASCADE
    ) ENGINE = InnoDB;

CREATETABLE ThreadDetails (
    Id INTPRIMARYKEY,
    PostCount INTNOTNULLDEFAULT0,
    LastModified DATE,
    LastMod VARCHAR(20),
        FOREIGNKEY (Id) REFERENCES Threads(Id)
        ONDELETECASCADE
        ONUPDATECASCADE
    ) ENGINE = InnoDB;

CREATETABLE Posts (
    Id INTPRIMARYKEY,
    Content VARCHAR(65000) NOTNULL,
    PostDate DATENOTNULL,
    Username VARCHAR(20) NOTNULL,
    ThreadId INTNOTNULL,
        FOREIGNKEY (Username) REFERENCES Users(Username)
        ONDELETECASCADE
        ONUPDATECASCADE,
        FOREIGNKEY (ThreadId) REFERENCES Threads(Id)
        ONDELETECASCADE
        ONUPDATECASCADE
    ) ENGINE = InnoDB;

CREATETABLE Conversations (
    Id INTPRIMARYKEY,
    Obj VARCHAR(100) NOTNULL
    ) ENGINE = InnoDB;

CREATETABLE UserConversation (
    Username VARCHAR(20) NOTNULL,
    Id INTNOTNULL,
        FOREIGNKEY (Username) REFERENCES Users(Username)
        ONDELETECASCADE
        ONUPDATECASCADE,
        FOREIGNKEY (Id) REFERENCES Conversations(Id)
        ONDELETECASCADE
        ONUPDATECASCADE
    ) ENGINE = InnoDB;

```

```

CREATETABLE Messages (
    Id INTPRIMARYKEY,
    Content VARCHAR(65000) NOTNULL,
    PostDate DATENOTNULL,
    Username VARCHAR(20) NOTNULL,
    Conversation INTNOTNULL,
        FOREIGNKEY (Username) REFERENCES Users(Username)
        ONDELETECASCADE
        ONUPDATECASCADE,
    FOREIGNKEY (Conversation) REFERENCES Conversations(Id)
        ONDELETECASCADE
        ONUPDATECASCADE
) ENGINE = InnoDB;

```

6 Query, Trigger e Funzioni

Di è riportato il codice SQL di alcune possibili interrogazioni effettuate alla base di dati. Le query non sono strettamente legate all'applicazione che fa uso del database.

Query 1: ritorna tutti i thread modificati da moderatori nella giornata corrente.

```

SELECT t.Title, m.Username
FROM ThreadDetails td NATURALJOIN Threads t, Moderators m
WHERE td.LastMod = m.Username AND td.LastModified IN (
    SELECT CURDATE()
)
GROUPBY t.Title

```

Output:

Title	Username
Nvidia maxwell skipping 20nm, jumps on 16nm!	micalleg
Sapphire Radeon HD 7950 one fan acting weird	micalleg
Unigine Valley Benchmark 1.0	micalleg

Query 2: Ritorna gli utenti che hanno aperto un thread in almeno due sezioni diverse

```

DROPVIEWIFEXISTS UserSections;
CREATEVIEW UserSections AS
SELECT t.Username, COUNT(t.SectionName) AS SectionsNumber
FROM Threads t
GROUPBY t.Username;

```

```

SELECT Username
FROM UserSections

```

WHERE SectionsNumber >1;

Output:

```
+-----+
| Username |
+-----+
| gcusinat |
| GoodBytes |
| micalleg |
| slickPC  |
+-----+
```

Query 3: Ritorna i threads con il numero massimo di post per ogni sezione

```
DROPVIEWIFEXISTS HelperView;
CREATEVIEW HelperView AS
SELECT t.Title, Count(p.Id) AS PostCount, t.SectionName
FROM Threads t JOIN Posts p ON (p.ThreadId = t.Id)
GROUPBY t.Id;
```

```
SELECT Title
FROM HelperView
GROUPBY SectionName
HAVINGMAX(PostCount);
```

Output:

```
+-----+
| Title |
+-----+
| LTT Forums CPU Overclocking Database! |
| Unigine Valley Benchmark 1.0 |
| Intel records strong Q2 earnings, beats PC market estimates |
+-----+
```

Query 4: Ritorna titolo, data e numero di post di tutti i thread aperti da moderatori con almeno due conversazioni attive

```
SELECT t.Title, t.PostDate, COUNT(p.Id) AS PostCount
FROM Threads t JOIN Posts p ON (p.ThreadId = t.Id)
WHERE t.Username IN (
    SELECT u.Username
    FROM Users u NATURALJOIN Moderators m
    WHERE2<= (
        SELECTCOUNT(*)
        FROM UserConversation uc
        WHERE uc.Username=u.Username
    )
)
GROUPBY t.Id;
```

Output:

Title	PostDate	PostCount
LTT Forums CPU Overclocking Database!	2014-07-16	1
AOC releases a 34inch 21:9 3440x1440 monitor	2014-07-16	1

Query 5: Ritorna i nomi degli utenti che hanno avuto almeno una conversazione con un admin

```
SELECT uc.Username
FROM UserConversation uc
WHERE uc.Id NOTIN (
    SELECT c.Id
    FROM UserConversation uc2 NATURALJOIN Conversations c
    WHERE uc2.Username NOTIN (SELECT *
        FROM Admins)
    AND c.Id NOTIN (SELECT uc3.Id
        FROM UserConversation uc3
        WHERE uc3.Username IN
            (
                SELECT *
                FROM Admins
            )
        )
    )
AND uc.Username NOTIN (SELECT * FROM Admins)
```

Output:

Username
micalleg

Query 6: Per ogni utente ritorna il numero di conversazioni a cui partecipa e il numero di messaggi inviati

```
SELECT uc.Username, COUNT(uc.Id) AS Conversations, COUNT(m.Id) as Messages
FROM UserConversation uc JOIN Messages m ON (uc.Username = m.Username)
GROUPBY m.Username
UNION
SELECT uc.Username, COUNT(uc.Id), COUNT(null)
FROM UserConversation uc
WHERE uc.Username NOTIN (
    SELECT m.Username
    FROM Messages m
)
GROUPBY uc.Username
UNION
SELECT u.Username, COUNT(null), COUNT(null)
FROM Users u
WHERE u.Username NOTIN (
    SELECT uc.Username
    FROM UserConversation uc
)
```

GROUPBY u.Username

Output:

Username	Conversations	Messages
micalleg	9	9
gcusinat	1	0
jedi12	1	0
slickPC	1	0
GoodBytes	0	0
pcEnt	0	0

I trigger (inclusi nel file forum.sql) mirano ad aggiornare il valore PostCount degli utenti non appena aggiungono un Post o nel caso lo stesso venga rimosso da un moderatore.

```
DROPTRIGGERIFEXISTS UpdatePostCount$$
CREATETRIGGER UpdatePostCount
AFTERINSERTON Posts
FOREACHROW
BEGIN
    DECLARE usr VARCHAR(20);
    DECLARE thread VARCHAR(20);
    SET usr = (SELECT u.Username
               FROM Users u
               WHERE u.Username = new.Username);
    SET thread = (SELECT p.ThreadId
                  FROM Posts p
                  WHERE p.Id = new.Id);
    IF usr ISNOTNULLTHEN
        UPDATE Users
        SET PostCount = PostCount + 1
        WHERE Username = usr;
    ENDIF;
    IF thread ISNOTNULLTHEN
        UPDATE ThreadDetails
        SET PostCount = PostCount + 1
        WHERE Id = thread;
    ENDIF;
END$$
```

```
DROPTRIGGERIFEXISTS DecreasePostCount$$
CREATETRIGGER DecreasePostCount
AFTERDELETEON Posts
FOREACHROW
BEGIN
    DECLARE usr VARCHAR(20);
    DECLARE thread VARCHAR(20);
    SET usr = (SELECT u.Username
               FROM Users u
               WHERE u.Username = old.Username);
    SET thread = (SELECT p.ThreadId
                  FROM Posts p
                  WHERE p.Id = old.Id);
```

```

IF usr ISNOTNULLTHEN
    UPDATE Users
    SET PostCount = PostCount - 1
    WHERE Username = usr;
ENDIF;
IF thread ISNOTNULLTHEN
    UPDATE ThreadDetails
    SET PostCount = PostCount - 1
    WHERE Id = thread;
ENDIF;
END$$

```

Queste invece sono le due funzioni (sempre incluse nel file forum.sql):

```

DROPFUNCTIONIFEXISTS MostActiveUser$$
CREATEFUNCTION MostActiveUser () RETURNSVARCHAR(20)
BEGIN
    DECLARE name VARCHAR(20);
    DECLARE pCount INT;
    SELECTmax(PostCount) INTO pCount FROM Users;
    SELECT Username INTO name FROM Users WHERE PostCount=pCount LIMIT1;
    RETURN name;
END$$

DROPFUNCTIONIFEXISTS HasPosted$$
CREATEFUNCTION HasPosted (usr VARCHAR(20)) RETURNSBOOLEAN
BEGIN
    DECLARE name VARCHAR(20);
    DECLARE posted BOOLEAN;
    SELECT Username INTO name
    FROM Posts
    WHERE Username = usr
    GROUPBY Username;
    IF name ISNOTNULL
        THEN
            SET posted = 1;
        ELSE
            SET posted = 0;
    ENDIF;
    RETURN posted;
END$$

DROPVIEWIFEXISTS UserSections;
CREATEVIEW UserSections AS
SELECT t.Username, COUNT(t.SectionName) AS SectionsNumber
FROM Threads t
GROUPBY t.Username;

SELECT Username
FROM UserSections
WHERE SectionsNumber >1;

```

6 Applicazione Web

Il file header.php contiene l'header dell'interfaccia web che sarà incluso un tutte le pagine del sito. È principalmente composto da un del codice per il controllo della sessione (quindi se l'utente è loggato o meno) e da un form per il login.

```

<divid="header">
<tableid="headtable">
<tr>
<td>
<a href="home.php?id="title"><strong>THETEK||
FORUM</strong></a>
</td>
<td align="right">
<?php
require "functions.php";
session_start();
$userinfo=$_SESSION['userinfo'];
if($userinfo)
{
$username=$userinfo['Username'];
$postcount=post_count($username);
echo"<br><table border='0' cellspacing='3' cellpadding='0'>
<tr></tr>
<tr>
<td class='option'>Welcome back, $username ($postcount
posts)</td>
</tr>
<tr>
<td>
<form action='processform.php?logout=true'
method='post' align=right>
<input type='submit' value='Log Out'>
</form>
</td>
</tr>
<tr>
<td><br></td>
</tr>
</table>";
}
else
{
?>
<form action='processform.php?login=true' method='post'>
<table border="0" cellspacing="3" cellpadding="0">
<tr>
<td class="formtext">UserName:</td>
<td><input type="text" name="username"></td>
</tr>
<tr>
<td class="formtext">Password:</td>
<td><input type="password" name="pwd"></td>
<td></td><td><input type="submit" value="Sign In">
</tr>
<tr>
<td>
<?php
if(isset($_GET['logerror']))
{
echo"<td colspan='2' style='color:red'>Incorrect username or password.</td>";
}
else
{
echo"<td colspan='2'>Username or password are incorrect.</td>";
}
}

```



```

?>
</tr>
</table>
</form>
<?php
}
?>
</td>
</tr>
<tr>
<td>
<a href="register.php" class="option"><strong>Register</strong></a>
<a href="messages.php" class="option"><strong>Messages</strong></a>
<a href="adminarea.php" class="option"><strong>AdminArea</strong></a>
</td>
<td></td>
</tr>
</table>
<!--/close main header table -->
</div>
<!-- /close header -->

```

Il file home.php contiene il codice per la home page del forum. Vengono effettuare delle interrogazioni al database per stampare in una tabella tutte le sezioni del forum.

```

<html>
<head>
<title>THETEK</title>
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
<?php require 'header.php'; ?>
<div id="content">
<table border="4" cellspacing="0" cellpadding="12">
<tr>
<td class="tablehead">Sections</td>
<td class="tablehead">Threads</td>
<td class="tablehead">Post</td>
</tr>
<?php
$conn=db_auth();
$sections=mysql_query("SELECT * FROM Sections WHERE SupSection='root'", $conn);
/*Scorro tutte le sezioni presenti nel db */
while($sec=mysql_fetch_assoc($sections))
{
$name=$sec['Name'];
$descr=$sec['Description'];
echo"<tr>
<td>
<a href='section.php?title=$name' class='section'><h3>$name</h3></a>
<font color='white'>$descr</font>";

/*Controllo se sono presenti sottosezioni */

```

```

$subSections=mysql_query("SELECT*FROM Sections WHERE SupSection!='root'", $conn);
while($subsec=mysql_fetch_assoc($subSections))
{
    $sname=$subsec['Name'];
    $sdescr=$subsec['Description'];
    $sup=$subsec['SupSection'];
    if($sup==$name)
    {
        echo"<span class='rectangle'></span><a href='section.php?title=$sname'
        class='subsection'>$sname</a>";
    }
}
echo"</td>";

/*Riempio le restanti righe, npost, nthread */
$total=mysql_query("SELECTCount(*) as Total FROM Threads WHERE
SectionName='$name'", $conn);
$row=mysql_fetch_assoc($total);
$nthread=$row['Total'];
echo"<td class='number'>$nthread</td>";

$total=mysql_query("SELECTCount(p.Id) as Total FROM Posts p, Threads t WHERE
t.SectionName='$name' AND p.ThreadId = t.Id", $conn);
$row=mysql_fetch_assoc($total);
$npost=$row['Total'];
echo"<td class='number'>$npost</td>";
}
?>
</tr>
</table>
</div>
<!--/close content container -->
</body>
</html>

```

Il file register.php contiene il form per la registrazione dell'utente al forum. Le elaborazioni di tutti i form avvengono nella pagina processform.php.

```

<html>
<head>
<title>REGISTER</title>
<linkhref="style.css"rel="stylesheet"type="text/css">
</head>
<body>
<?phprequire'header.php';?>
<divid="content">
<formname='modulo'action='processform.php?reg=true'method='post'>
<tableborder="0"cellspacing="5"cellpadding="5">
<tr>
<tdclass="formtext">Username:</td><td><inputtype="text"name="username"></td>
</tr>
<tr>
<tdclass="formtext">Email:</td><td><inputtype="text"name="email"></td>
</tr>

```

```

<tr>
<tdclass="formtext">Password:</td><td><inputtype="password"name="pwd1"></td>
</tr>
<tr>
<tdclass="formtext">ConfirmPassword:</td><td><inputtype="password"name="pwd2"></td>
</tr>
<tr>
<td></td><td><inputtype="submit"value="Confirm">
<inputtype="reset"value="Reset"></td></tr></td>
</tr>
<table>
</form>
<div>
<!--/close content div -->
<?php
if($_GET)
{
$m=$_GET['m'];
switch($m)
{
case"empty":echo"Some filels are empty. Fill out the form and try again.";
break;
case"password":echo"The password fields did not match.";
break;
case"username":echo"The username is already in use.";
break;
case"email":echo"The email address is already in use.";
break;
}
}
?>
</body>
</html>

```

Il file processform.php elabora tutte le informazioni proveniente dai vari form delle altre pagine php. Il form da processare viene identificato tramite il superglobal array \$_GET che contiene l'identificatore della pagina da cui proviene. I dati vengono raccolti tramite il classico array \$_POST ed eventualmente aggiunti al database. In caso di errore o fine dell'elaborazione, l'utente viene indirizzato alla pagina richiesta tramite la funzione header(..) (utilizzabile visto che processform.php non emette nessun output).

```

<?php
require'functions.php';

/*Url corretto del server per il redirecting */
$server_url=server_url();

/*Autenticazione al database */
$conn=db_auth();

/*Inizio la sessione per verificare i dati dell'utente */
session_start();

/*Controllo logout */

```

```

if(isset($_GET['logout']))
{
unset($_SESSION['userinfo']);
header("Location:".$_SERVER['HTTP_REFERER']);
}
/*Controllo login */
elseif(isset($_GET['login']))
{
$password=$_POST['pwd'];
$username=$_POST['username'];

/*Interrogo il database e controllo l'esistenza dell'utente */
$user=mysql_query("SELECT*FROM Users WHERE Username='$username'", $conn)
or die("Authentication failed. ".mysql_error($conn));

if($row=mysql_fetch_assoc($user))
{
/*L'utente esiste, controllo la password */
if($password==$row['Pwd'])
{
/*Dati corretti, autentico l'utente */
login($row);
header("Location:".$_SERVER['HTTP_REFERER']);
exit;
}
else
{
/*Password errata */
header("Location:".$server_url."home.php?logerror=true");
exit;
}
}
else
{
/*L'utente non esiste */
header("Location:".$server_url."home.php?logerror=true");
exit;
}

}
/*Controllo registrazione */
elseif(isset($_GET['reg']))
{
/*Ottieni i valori in POST */
$password=$_POST['pwd1'];
$passwordCheck=$_POST['pwd2'];
$username=$_POST['username'];
$email=$_POST['email'];

/*Controlla che i campi non siano vuoti */
if($password=="" || $username=="" || $email=="")
{
header("Location:".$server_url."register.php?m=empty");
exit;
}

/*Controllo della password */

```

```

if($password!=$passwordCheck)
{
header("Location:".$server_url."register.php?m=password");
exit;
}
/*Controllo unicità username ed email */
$users=mysql_query("SELECT*FROM Users",$conn)
ordie("Operation failed.".mysql_error($conn));
while($row=mysql_fetch_assoc($users))
{
if($username==$row['Username'])
{
header("Location:".$server_url."register.php?m=username");
exit;
}
if($email==$row['Email'])
{
header("Location:".$server_url."register.php?m=email");
exit;
}
}
/*Aggiungo l'utente al database e lo autentico */
save_user($username,$password,$email,$conn);
header("Location:".$server_url."home.php");
exit;
}
/*Processo il nuovo thread */
elseif(isset($_GET['newthread']))
{
$section=$_GET['sec'];
/*Se il titolo risulta vuoto, errore */
if(empty($_POST['title']))
{
header("Location:".$server_url."newthread.php?sec=$section&empty=true");
exit;
}
$title=mysql_real_escape_string($_POST['title']);
$text=mysql_real_escape_string($_POST['text']);
add_thread($title,$text,$section,$conn);
header("Location:".$server_url."section.php?title=$section");
exit;
}
/*Processo il nuovo post */
elseif(isset($_GET['newpost']))
{
$text=mysql_real_escape_string($_POST['text']);
$id=$_GET['id'];
/*Se il testo risulta vuoto, errore */
if(empty($_POST['text']))
{
header("Location:".$server_url."thread.php?id=$id&empty=true");
exit;
}
add_post($text,$id,$conn);
header("Location:".$server_url."thread.php?id=$id");
exit;
}

```

```

/*Processo la nuova conversazione */
elseif(isset($_GET['newconv']))
{
$receivers_str=$_POST['receivers'];
$receivers=explode(",",$receivers_str);
if(empty($receivers)||!receivers_exists($receivers,$conn))
{
header("Location:".$server_url."newconversation.php?receivers=true");
exit;
}
$obj=mysql_real_escape_string($_POST['object']);
$text=mysql_real_escape_string($_POST['text']);
if(empty($obj)||empty($text))
{
header("Location:".$server_url."newconversation.php?empty=true");
exit;
}

$id=add_conversation($obj,$receivers,$conn);
add_message($text,$id,$conn);
header("Location:".$server_url."conversation.php?id=$id");
exit;
}
/*Processo il nuovo messaggio */
elseif(isset($_GET['newmess']))
{
$text=mysql_real_escape_string($_POST['text']);
$id=$_GET['id'];
add_message($text,$id,$conn);
header("Location:".$server_url."conversation.php?id=$id");
exit;
}
/*Inizializza l'update o delete del thread */
elseif(isset($_GET['managethread']))
{
$id=$_GET['id'];
$sec=get_section($id,$conn);
if($_POST['action']=='Delete')
{
mysql_query("DELETEFROM Threads WHERE Id='$id'", $conn);
header("Location:".$server_url."section.php?title=$sec&id=$id");
exit;
}
else
{
header("Location:".$server_url."edit.php?thread=$id");
exit;
}
}
/*Processo il thread modificato */
elseif(isset($_GET['modthread']))
{
$id=$_GET['id'];
$sec=get_section($id,$conn);
/*Se il titolo risulta vuoto, errore */
if(empty($_POST['title']))
{

```

```

header("Location:".$server_url."edit.php?thread=$id&empty=true");
exit;
}
$title=mysql_real_escape_string($_POST['title']);
$text=mysql_real_escape_string($_POST['text']);
update_thread($title,$text,$id,$conn);
header("Location:".$server_url."thread.php?id=$id");
exit;
}
/*Inizializza l'update o delete del post */
elseif(isset($_GET['managepost']))
{
$id=$_GET['id'];
$thread_id=get_thread($id,$conn);
if($_POST['action']=='Delete')
{
mysql_query("DELETEFROM Posts WHERE Id='$id'", $conn);
header("Location:".$server_url."thread.php?id=$thread_id");
exit;
}
else
{
header("Location:".$server_url."edit.php?post=$id");
exit;
}
}
/*Processo il post modificato */
elseif(isset($_GET['modpost']))
{
$id=$_GET['id'];
$thread_id=get_thread($id,$conn);
/*Se il titolo risulta vuoto, errore */
if(empty($_POST['text']))
{
header("Location:".$server_url."edit.php?post=$id&empty=true");
exit;
}
$text=mysql_real_escape_string($_POST['text']);
mysql_query("UPDATE Posts SET Content='$text' WHERE Id='$id'", $conn);
header("Location:".$server_url."thread.php?id=$thread_id");
exit;
}
/*Inizializzo update o delete di sezioni ed utenti, Admin area */
elseif(isset($_GET['adminarea']))
{
$name=$_POST['name'];
if($_POST['action']=='delete')
{
if($_POST['object']=='section')
{
/*Delete sezione */
if(is_section($name,$conn))
{
mysql_query("DELETEFROM Sections WHERE Name='$name'", $conn);
header("Location:".$server_url."adminarea.php?secdel=true");
exit;
}
}
}
}

```

```

else
{
header("Location:".$server_url."adminarea.php?nosec=true");
exit;
}
}
else
{
/*Delete user */
if(is_user($name,$conn))
{
mysql_query("DELETEFROM Users WHERE Username='$name'", $conn);
header("Location:".$server_url."adminarea.php?usrdel=true");
exit;
}
else
{
header("Location:".$server_url."adminarea.php?nouser=true");
exit;
}
}
}
else
{
if($_POST['object']=='section')
{
/*Update sezione */
if(is_section($name,$conn))
{
header("Location:".$server_url."edit.php?section=$name");
exit;
}
else
{
header("Location:".$server_url."adminarea.php?nosec=true");
exit;
}
}
else
{
/*Update user */
if(is_user($name,$conn))
{
header("Location:".$server_url."edit.php?user=$name");
exit;
}
else
{
header("Location:".$server_url."adminarea.php?nouser=true");
exit;
}
}
}
}
/*Processo la sezione modificata */
elseif(isset($_GET['modsec']))
{

```



```

$name=$_GET['name'];
/*Se il titolo risulta vuoto, errore */
if(empty($_POST['title']))
{
header("Location:".$server_url."edit.php?section=$name&empty=true");
exit;
}
$title=mysql_real_escape_string($_POST['title']);
$description=mysql_real_escape_string($_POST['description']);
mysql_query("UPDATE Sections SET Name='$title', Description='$description' WHERE
Name='$name'", $conn);
header("Location:".$server_url."home.php");
exit;
}
/*Processo l'username modificato */
elseif(isset($_GET['modusr']))
{
$name=$_GET['username'];
/*Se vuoto, errore */
if(empty($_POST['username']))
{
header("Location:".$server_url."edit.php?user=$name&empty=true");
exit;
}
$username=$_POST['username'];
/*Controllo se esiste gia' */
$user=mysql_query("SELECT*FROM Users WHERE Username='$username'", $conn)
or die("Operation failed.".mysql_error($conn));
if($row=mysql_fetch_assoc($user))
{
if($username==$row['Username'])
{
header("Location:".$server_url."edit.php?user=$name&exist=true");
exit;
}
}
else
{
mysql_query("UPDATE Users SET Username='$username' WHERE Username='$name'", $conn);
header("Location:".$server_url."adminarea.php?usrup=true");
exit;
}
}
?>

```

Il file newthread.php contiene il form che consente ad un utente di aggiungere un thread in una data sezione.

```

<head>
<title>THETEK</title>
<linkhref="style.css"rel="stylesheet"type="text/css">
</head>
<body>
<?phprequire'header.php';?>
<divid="content">

```

```

<?php
$section=$_GET['sec'];
/*Devo aggiungere un thread a l db */
echo"<form action='processform.php?newthread=true&sec=$section' method='post'>
<pclass='areatitle'>Title:</p><br>
<inputtype='text' name='title' style='width:57%'><br>
<pclass='areatitle'>Text:</p><br>
<textareaname='text' rows='24' cols='120'></textarea><br>";

/*Controllo se l'utente e' loggato */
if(is_logged())
{
echo"<input type='submit' value='Publish' style='margin:12px 0px 0px 0px' /><br>
</form>";
}
else
{
echo"<p class='formtext'><a href='register.php' style='color:white'>Register</a> or
login in order to post new content in the forum</p>
</form>";
}

if(isset($_GET['empty']))
{
echo"<p style='color:red'>TITLE CAN'T BE EMPTY</p>";
}
?>

</div>
</body>
</html>

```

Il file newconversation.php contiene il form per creare una nuova conversazione con i dovuti controlli (es. controllo esistenza destinatari).

```

<head>
<title>THETEK</title>
<linkhref="style.css"rel="stylesheet"type="text/css">
</head>
<body>
<?phprequire'header.php';?>
<divid="content">
<?php
echo"<form action='processform.php?newconv=true' method='post'>"
?>
<pclass="areatitle">Receivers(separatedbycommas):</p><br>
<inputtype="text" name="receivers" style="width:57%"><br>
<pclass="areatitle">Object:</p><br>
<inputtype="text" name="object" style="width:57%"><br>
<pclass="areatitle">Text:</p><br>
<textareaname="text" rows="24" cols="120"></textarea><br>
<?php
/*Controllo se l'utente e' loggato */
if(is_logged())

```

```

{
echo"<input type='submit' value='Send' style='margin:12px 0px 0px 0px' /><br>
</form>";
}
else
{
echo"<p class='formtext'><a href='register.php' style='color:white'>Register</a> or
login in order to access your conversations.</p>
</form>";
}

if(isset($_GET['empty']))
{
echo"<p style='color:red'>SOME FILES ARE EMPTY. PLEASE; FILL THE FORM AND TRY
AGAIN</p>";
}
if(isset($_GET['receivers']))
{
echo"<p style='color:red'>RECEIVERS FIELD IS NOT VALID</p>";
}
?>

</div>
<!--/close content container -->
</body>
</html>

```

Il file messages.php contiene l'elenco di tutte le conversazioni aperte di un utente.

```

<html>
<head>
<title>THETEK</title>
<linkhref="style.css"rel="stylesheet"type="text/css">
</head>
<body>
<?phprequire'header.php';?>
<divid="content">
<?php
if(!is_logged())
{
echo"<p class='formtext'><a href='register.php' style='color:white'>Register</a> or
login in order to access your conversations.</p>";
exit;
}
else
{
/*Header numero conversazioni */
$conn=db_auth();
$userinfo=$_SESSION['userinfo'];
$username=$userinfo['Username'];
$conversations=mysql_query("SELECTCount(Id) as Total FROM UserConversation WHERE
Username = '$username'", $conn);
$row=mysql_fetch_assoc($conversations);
$count=$row['Total'];

```

```

if($count!=1)
echo"<h3 class='responses'>$count CONVERSATIONS</h3>";
else
echo"<h3 class='responses'>1 CONVERSATION</h3>";
}
?>
<tableborder='4'cellspacing='0'cellpadding='12'>
<tr>
<td>
<?php
echo"<a href='newconversation.php' class='section'>NEW CONVERSATION...</a>";
?>
</td>
</tr>
<?php
/*Visualizzo le conversazioni */
$conn=db_auth();
$userinfo=$_SESSION['userinfo'];
$username=$userinfo['Username'];
$conversations=mysql_query("SELECT c.Id, c.Obj
FROMConversationscNATURALJOINUserConversationu
WHEREu.Username='$username'GROUPBYc.Id",
$conn);
while($row=mysql_fetch_assoc($conversations))
{
$obj=strtoupper($row['Obj']);
$id=$row['Id'];
echo"<tr>
<td><a href='conversation.php?id=$id' class='section'><h4>$obj<h4><a></td>
</tr>";
}
?>
</table>
</div>
<!--/close content container -->
</body>
</html>

```

Il file functions.php contiene una serie di funzioni che mirano principalmente ad interrogare il database. Risulta incluso in tutte le altre pagine. Sono presenti, ad esempio, funzioni per l'autenticazione al database, per la creazione di una sessione tramite \$_SESSION, e di aggiunta dei vali valori nella base di dati.

```

<?php
/*Ritorna l'attuale url di base per il server */
functionserver_url()
{
$server=$_SERVER["SERVER_NAME"];
if($server=="localhost")
{
return"http://localhost:20080/basidati/~gcusinat/";
}
else//($server == "basidati")
{
return"http://basidati/basidati/~gcusinat/";
}
}

```

```

}
}

/*Inizia una sessione per autenticare l'utente */
functionlogin($userinfo)
{
$_SESSION['userinfo']=$userinfo;
}

/*Autenticazione al database */
functiondb_auth()
{
$host="basidati1004.studenti.math.unipd.it";
$user="gcusinat";
$pwd="LzdSOz8H";
$conn=mysql_connect($host,$user,$pwd)
ordie($_SERVER['PHP_SELF']."Connection failed, please try again!");
mysql_select_db("gcusinat-PR",$conn)
ordie("Could not select gcusinat-PR");
return$conn;
}

/*Aggiungo l'utente al database e lo autentico */
functionsave_user($username,$password,$email,$conn)
{
$date=newDateTime();
$today=$date->format('Y-m-d');
$insert_query="INSERT INTO Users (Username, Pwd, Email, RegDate, PostCount)
VALUES ('$username', '$password', '$email', '$today', 0)";
mysql_query($insert_query,$conn)
ordie("Error in adding new user. ".mysql_error($conn));
mysql_close($conn);
$userinfo=array(
"Username"=>$username,
"Pwd"=>$password,
"Email"=>$email,
"RegDate"=>$today,
"PostCount"=>0
);
login($userinfo);
}

/*Controllo se l'utente e' loggato */
functionis_logged()
{
if(isset($_SESSION['userinfo']))
returntrue;
else
returnfalse;
}

/*Aggiungo un thread ad una sezione */
functionadd_thread($title,$text,$section,$conn)
{
$nthread=mysql_query("SELECTMAX(Id) as Count FROM Threads",$conn)
ordie("Error in creating the thread. ".mysql_error($conn));
$row=mysql_fetch_assoc($nthread);

```

```

$id=$row['Count']+1;

$userinfo=$_SESSION['userinfo'];
$username=$userinfo['Username'];

$date=newDateTime();
$today=$date->format('Y-m-d');

$insert_query="INSERT INTO Threads (Id, Title, Content, PostDate, Username,
SectionName)
VALUES ($id, '$title', '$text', '$today', '$username', '$section')";
mysql_query($insert_query,$conn)
ordie("Error in creating the thread. ".mysql_error($conn));
mysql_close($conn);
}

/*Aggiungo un post ad un thread */
functionadd_post($text,$thread_id,$conn)
{
$npost=mysql_query("SELECTMAX(Id) as Count FROM Posts",$conn)
ordie("Error in creating the post. ".mysql_error($conn));
$row=mysql_fetch_assoc($npost);
$id=$row['Count']+1;

$userinfo=$_SESSION['userinfo'];
$username=$userinfo['Username'];

$date=newDateTime();
$today=$date->format('Y-m-d');

$insert_query="INSERT INTO Posts (Id, Content, PostDate, Username, ThreadId)
VALUES ($id, '$text', '$today', '$username', '$thread_id')";
mysql_query($insert_query,$conn)
ordie("Error in creating the post. ".mysql_error($conn));
mysql_close($conn);
}

/*Aggiungo una conversazione, ritorno l'id per il messaggio iniziale */
functionadd_conversation($obj,$receivers,$conn)
{
$nconv=mysql_query("SELECTMAX(Id) as Count FROM Conversations",$conn)
ordie("Error in creating the post. ".mysql_error($conn));
$row=mysql_fetch_assoc($nconv);
$id=$row['Count']+1;

mysql_query("INSERT INTO Conversations (Id, Obj) VALUES ($id, '$obj')",$conn)
ordie("Error in creating the conversation. ".mysql_error($conn));

/*Aggiungo i valori alla tabella UserConversation */
$userinfo=$_SESSION['userinfo'];
$username=$userinfo['Username'];

mysql_query("INSERT INTO UserConversation (Username, Id) VALUES ('$username',
'$id')",$conn)
ordie("Error in creating the conversation. ".mysql_error($conn));

foreach($receiversas$value)

```

```

{
mysql_query("INSERT INTO UserConversation (Username, Id) VALUES ('$value',
'$id')", $conn)
ordie("Error in creating the conversation. ".mysql_error($conn));
}

return $id;
}

/*Aggiungo un messaggio alla conversazione */
function add_message($content, $convid, $conn)
{
    $nmess=mysql_query("SELECT MAX(Id) as Count FROM Messages", $conn)
    ordie("Error in creating the post. ".mysql_error($conn));
    $row=mysql_fetch_assoc($nmess);
    $id=$row['Count']+1;

    $userinfo=$_SESSION['userinfo'];
    $username=$userinfo['Username'];

    $date=new DateTime();
    $today=$date->format('Y-m-d');

    mysql_query("INSERT INTO Messages (Id, Content, PostDate, Username, Conversation)
VALUES ($id, '$content', '$today', '$username', '$convid')", $conn)
    ordie("Error in creating the message. ".mysql_error($conn));
}

/*Controlla se i mittenti esistono */
function receivers_exists($receivers, $conn)
{
    foreach($receivers as $value)
    {
        $user=mysql_query("SELECT * FROM Users WHERE Username='$value'", $conn)
        ordie("Operation failed. ".mysql_error($conn));
        if($row=mysql_fetch_assoc($user))
        {
            //found
        }
        else
        {
            return false;
        }
    }
    return true;
}

/*Controlla se l'utente loggato e' un mod */
function is_mod()
{
    $userinfo=$_SESSION['userinfo'];
    $username=$userinfo['Username'];

    $conn=db_auth();

    $mod=mysql_query("SELECT * FROM Moderators WHERE Username='$username'", $conn);
    if($row=mysql_fetch_assoc($mod))

```

```

{
returntrue;
}
else
{
$admin=mysql_query("SELECT*FROM Admins WHERE Username='$username'", $conn);
if($row=mysql_fetch_assoc($admin))
{
returntrue;
}
else
{
returnfalse;
}
}
}

/*Controlla se l'utente loggato e' un admin */
functionis_admin()
{
if(!is_logged())
{
returnfalse;
}

$userinfo=$_SESSION['userinfo'];
$username=$userinfo['Username'];

$conn=db_auth();

$admin=mysql_query("SELECT*FROM Admins WHERE Username='$username'", $conn);
if($row=mysql_fetch_assoc($admin))
{
returntrue;
}
else
{
returnfalse;
}
}

/*Ritorna il nome della sezione dato l-id del thread */
functionget_section($id, $conn)
{
$section=mysql_query("SELECT*FROM Threads WHERE Id=$id", $conn);
if($row=mysql_fetch_assoc($section))
{
$name=$row['SectionName'];
return$name;
}
}

/*Aggiorna i valori di un thread */
functionupdate_thread($title, $text, $id, $conn)
{
mysql_query("UPDATE Threads SET Title='$title', Content='$text' WHERE
Id='$id'", $conn);
}

```



```

}

/*Ottiene il thread padre del post */
functionget_thread($id,$conn)
{
$thread=mysql_query("SELECT*FROM Posts WHERE Id=$id",$conn);
if($row=mysql_fetch_assoc($thread))
{
$thread_id=$row['ThreadId'];
return$thread_id;
}
}

/*Controlla se esiste una sezione con un dato nome */
functionis_section($name,$conn)
{
$section=mysql_query("SELECT*FROM Sections WHERE Name='$name'", $conn);
if($row=mysql_fetch_assoc($section))
{
if($row['Name']==$name)
returntrue;
else
returnfalse;
}
returnfalse;
}

/*Controlla se esiste un utente con un dato username */
functionis_user($name,$conn)
{
$user=mysql_query("SELECT*FROM Users WHERE Username='$name'", $conn);
if($row=mysql_fetch_assoc($user))
{
if($row['Username']==$name)
returntrue;
else
returnfalse;
}
returnfalse;
}

/*Ottiene il post count di un utente */
functionpost_count($username)
{
$conn=db_auth();
$user=mysql_query("SELECT*FROM Users WHERE Username='$username'", $conn);
if($row=mysql_fetch_assoc($user))
{
$count=$row['PostCount'];
return$count;
}
returnfalse;
}
?>

```

Il file edit.php consente agli admin o ai moderatori di editare un post, un thread, una sezione o un user (gli ultimi due solo per l'admin) con un form differente per ogni opzione.

```
<html>
<head>
<title>THETEK</title>
<linkhref="style.css"rel="stylesheet"type="text/css">
</head>
<body>
<?phprequire'header.php';?>
<divid="content">
<?php
/*Controllo se devo modificare thread, post, sezione o user */
if(isset($_GET['thread']))
{
$id=$_GET['thread'];
$conn=db_auth();
$thread=mysql_query("SELECT*FROM Threads WHERE Id=$id",$conn);
if($row=mysql_fetch_assoc($thread))
{
$content=$row['Content'];
$title=$row['Title'];
echo"<form action='processform.php?modthread=true&id=$id' method='post'>
<pclass='areatitle'>EditTitle:</p><br>
<inputtype='text' name='title' value='$title' style='width:57%'><br>
<pclass='areatitle'>EditText:</p><br>
<textareaname='text' rows='24' cols='120'>$content</textarea><br>
<inputtype='submit' value='Update' style='margin:12px 0px 0px 0px' /><br>
</form>";
if(isset($_GET['empty']))
{
echo"<p style='color:red'>EMPTY FIELDS DETECTED!</p>";
}
}
}
elseif(isset($_GET['post']))
{
$id=$_GET['post'];
$conn=db_auth();
$post=mysql_query("SELECT*FROM Posts WHERE Id='$id'", $conn);
if($row=mysql_fetch_assoc($post))
{
$content=$row['Content'];
echo"<form action='processform.php?modpost=true&id=$id' method='post'>
<pclass='areatitle'>EditText:</p><br>
<textareaname='text' rows='24' cols='120'>$content</textarea><br>
<inputtype='submit' value='Update' style='margin:12px 0px 0px 0px' /><br>
</form>";
if(isset($_GET['empty']))
{
echo"<p style='color:red'>CONTENT CAN'T BE EMPTY!</p>";
}
}
}
elseif(isset($_GET['section']))
```

```

{
$name=$_GET['section'];
$conn=db_auth();
$section=mysql_query("SELECT*FROM Sections WHERE Name='$name'", $conn);
if($row=mysql_fetch_assoc($section))
{
$description=$row['Description'];
echo"<form action='processform.php?modsec=true&name=$name' method='post'>
<pclass='areatitle'>EditName:</p><br>
<inputtype='text' name='title' value='$name' style='width:57%'><br>
<pclass='areatitle'>EditDescription:</p><br>
<textarea name='description' rows='24' cols='120'>$description</textarea><br>
<inputtype='submit' value='Update' style='margin:12px 0px 0px 0px'><br>
</form>";
if(isset($_GET['empty']))
{
echo"<p style='color:red'>EMPTY FIELDS DETECTED!</p>";
}
}
elseif(isset($_GET['user']))
{
$username=$_GET['user'];

echo"<form action='processform.php?modusr=true&username=$username' method='post'>
<pclass='areatitle'>Editusername:</p><br>
<inputtype='text' name='username' value='$username' style='width:57%'><br>
<inputtype='submit' value='Update' style='margin:12px 0px 0px 0px'><br>
</form>";
if(isset($_GET['empty']))
{
echo"<p style='color:red'>EMPTY FIELDS DETECTED!</p>";
}
if(isset($_GET['exist']))
{
echo"<p style='color:red'>USERNAME ALREADY EXISTS!</p>";
}
}
?>
</div>
</body>
</html>

```

Il file conversation.php visualizza il contenuto di una conversazione con tutti i messaggi contenuti in essa rappresentati in una tabella.

```

<html>
<head>
<title>THETEK</title>
<linkhref="style.css"rel="stylesheet"type="text/css">
</head>
<body>
<?phpinclude"header.php"?>
<divid='content'>

```

```

<?php
/*Stampo l'oggetto della conversazione */
$id=$_GET['id'];
$conn=db_auth();
$conv=mysql_query("SELECT*FROM Conversations WHERE Id='$id'", $conn);
if($row=mysql_fetch_assoc($conv))
{
$obj=strtoupper($row['Obj']);
echo"<h3 class='responses'>$obj</h3>";
}

/*Stampo i partecipanti */
echo"<p class='thread'>Participants: ";
$users=mysql_query("SELECT u.Username FROM UserConversation uc NATURAL JOIN Users u
WHERE uc.Id=$id", $conn);
if($row=mysql_fetch_assoc($users))
{
$usr=$row['Username'];
echo"$usr";
}
while($row=mysql_fetch_assoc($users))
{
$usr=$row['Username'];
echo", $usr";
}
echo"</p>";

/*Lista messaggi */
$mess=mysql_query("SELECT*FROM Messages WHERE Conversation='$id'", $conn);
echo"<table border='2' cellpadding='12'>";
while($row=mysql_fetch_assoc($mess))
{
$content=$row['Content'];
$postdate=$row['PostDate'];
$username=$row['Username'];

echo"<tr>
<td>
<pstyle='color:white'>$content</p>
<pclass='thread'>postedby$usernameon$postdate</p>
</td>
</tr>";
}
echo"</table>";

/*Form per nuovo messaggio */
echo"<form action='processform.php?newmess=true&id=$id' method='post'>
<pclass='areatitle'>Text:</p><br>
<textareaname='text'rows='24'cols='120'></textarea><br>";
if(is_logged())
{
echo"<input type='submit' value='Post' style='margin:12px 0px 0px 0px' /><br>
</form>";
}
else
{

```

```

echo"<p class='formtext'><a href='register.php' style='color:white'>Register</a> or
login in order to post new content in the forum</p>
</form>";
}
if(isset($_GET['empty']))
{
echo"<p style='color:red'>MESSAGE CAN'T BE EMPTY</p>";
}
?>

<div>
<!--/close div content tag -->
</body>
</html>

```

Il file adminarea.php permette all'admin di scegliere se modificare un utente o una sezione.

```

<html>
<head>
<title>THETEK</title>
<linkhref="style.css"rel="stylesheet"type="text/css">
</head>
<body>
<?phpinclude"header.php"?>
<divid='content'>
<?php
/*Solo gli admin possono accedere a quest'area */
if(!is_admin())
{
echo"<p style='font-family:helvetica'>Restricted area. Only admins have access to
this page.</p>";
exit;
}
?>
<formaction='processform.php?adminarea=true'method='post'>
<h3>Action:</h3>
<inputtype="radio"name="action"value="update">Update
<inputtype="radio"name="action"value="delete">Delete
<h3>Object:</h3>
<inputtype="radio"name="object"value="section">Section
<inputtype="radio"name="object"value="user">User
<p></p>
Section/User name:<input type="text" name="name">
<p></p>
<inputtype="submit"value="Confirm">
</form>
<?php
if(isset($_GET['nosec']))
{
echo"<p style='color:red'>ENTERED SECTION DOES NOT EXIST. PLEASE, TRY AGAIN</p>";
}
if(isset($_GET['secdel']))
{
echo"<p style='color:red'>SECTION DELETED SUCCESSFULLY!</p>";
}
}

```

```

}
if(isset($_GET['nouser']))
{
echo"<p style='color:red'>ENTERED USER DOES NOT EXIST. PLEASE, TRY AGAIN</p>";
}
if(isset($_GET['usrdel']))
{
echo"<p style='color:red'>USER DELETED SUCCESSFULLY!</p>";
}
if(isset($_GET['usrup']))
{
echo"<p style='color:red'>USERNAME UPDATED SUCCESSFULLY!</p>";
}
?>
<div>
</body>
</html>

```

Il file thread.php contiene la visualizzazione di un dato thread con i relativi post. In caso la sessione aperta identifica un mod o un admin, vengono aggiunti le opzioni per modificare o eliminare ogni post.

```

<html>
<head>
<title>THETEK</title>
<linkhref="style.css"rel="stylesheet"type="text/css">
</head>
<body>
<?phpinclude"header.php"?>
<divid='content'>
<?php
/*Creo il tread */
$id=$_GET['id'];
$conn=db_auth();
$thread=mysql_query("SELECT*FROM Threads WHERE Id='$id'", $conn);
if($row=mysql_fetch_assoc($thread))
{
$title=strtoupper($row['Title']);
$content=$row['Content'];
$postdate=$row['PostDate'];
$username=$row['Username'];

echo"<table border='2' cellpadding='12'>
<tr>
<td>
<h2class='thread'>$title</h2>
<pstyle='color:white'>$content</p>
<pclass='thread'>postedby$usernameon$postdate</p>";

if(is_logged() && is_mod())
echo"<form action='processform.php?managethread=true&id=$id' method='post'>
<inputtype='submit'value='Delete'name='action'/>
<inputtype='submit'value='Update'name='action'/>
</form>";

```

```

echo"</td>
</tr>
</table>";
}

/*Label il numero di post presenti */
$postts=mysql_query("SELECTCount(Id) as Total FROM Posts WHERE
ThreadId='$id'", $conn);
if($row=mysql_fetch_assoc($posts))
{
$count=$row['Total'];
echo"<h3 class='responses'>$count RESPONSES TO THIS THREAD</h3>";
}
else
{
echo"<h3 class='responses'>0 RESPONSES TO THIS THREAD</h3>";
}

/*Lista posts */
$postts=mysql_query("SELECT*FROM Posts WHERE ThreadId='$id'", $conn);
echo"<table border='2' cellpadding='12'>";
while($row=mysql_fetch_assoc($posts))
{
$content=$row['Content'];
$postdate=$row['PostDate'];
$username=$row['Username'];
$post_id=$row['Id'];

echo"<tr>
<td>
<pstyle='color:white'>$content</p>
<pclass='thread'>postedby$usernameon$postdate</p>";
if(is_logged() &&is_mod())
{
echo"<form action='processform.php?managepost=true&id=$post_id' method='post'>
<inputtype='submit' value='Delete' name='action' />
<inputtype='submit' value='Update' name='action' />
</form>";
}
echo"</td>
</tr>";
}
echo"</table>";

/*Form per nuovo post */
echo"<form action='processform.php?newpost=true&id=$id' method='post'>
<pclass='areatitle'>Text:</p><br>
<textareaname='text' rows='24' cols='120'></textarea><br>";
if(is_logged())
{
echo"<input type='submit' value='Post' style='margin:12px 0px 0px 0px' /><br>
</form>";
}
else
{
echo"<p class='formtext'><a href='register.php' style='color:white'>Register</a> or
login in order to post new content in the forum</p>";
}

```

```

</form>";
}
if(isset($_GET['empty']))
{
echo"<p style='color:red'>TITLE CAN'T BE EMPTY</p>";
}
?>

<div>
<!--/close div content tag -->
</body>
</html>

```

Il file section.php rappresenta una sezione con l'elenco dei thread contenuti in essa. Risulta inoltre presente un hyperlink per aggiungere un nuovo thread.

```

<html>
<head>
<title>THETEK</title>
<linkhref="style.css"rel="stylesheet"type="text/css">
</head>
<body>
<?phprequire'header.php';?>
<divid="content">
<tableborder='4'cellspacing='0'cellpadding='12'>
<tr>
<?php
$conn=db_auth();
$title=$_GET['title'];
$section=mysql_query("SELECT*FROM Sections WHERE Name='$title'", $conn);
if($sec=mysql_fetch_assoc($section))
{
$descr=$sec['Description'];
echo"<td>
<h2class='section'>$title</h2>
<fontcolor='white'>$descr</font>";
/*Controllo se sono presenti sottosezioni */
$subSections=mysql_query("SELECT*FROM Sections WHERE SupSection='$title'", $conn);
while($subsec=mysql_fetch_assoc($subSections))
{
$sname=$subsec['Name'];
$sdescr=$subsec['Description'];
echo"<span class='rectangle'></span><a href='section.php?title=$sname'
class='subsection'>$sname</a>";
}
}
?>
</td>
<tr>
<td>
<?php
$section=$_GET['title'];
echo"<a href='newthread.php?sec=$title' class='section'>NEW THREAD...</a>";
?>

```



```

</td>
</tr>
<?php
/*Visualizzo i thread */
$conn=db_auth();
$section=$_GET['title'];
$thread=mysql_query("SELECT t.Id, t.Title FROM Threads t WHERE
t.SectionName='$section' ORDER BY t.Id DESC", $conn);
while($row=mysql_fetch_assoc($thread))
{
$title=strtoupper($row['Title']);
$id=$row['Id'];
echo"<tr>
<td><a href='thread.php?id=$id' class='section'><h4>$title<h4><a></td>
</tr>";
}
?>
</table>
</div>
<!--/close content container -->
</body>
</html>

```