

Database application: Documentation

Introduction

This app is for a discussion forum database application. The key function of the app is to have users being able to create and read discussion threads and make posts in them for others to read.

The app is written in PHP using PostgreSQL as the database. The application runs at the users.cs.helsinki.fi server of the Department of Computer Science of University of Helsinki.

Use instructions

The application is usable in web address *<http://tuomokar.users.cs.helsinki.fi/tsoha/>*.

User can login to an admin account with:

- Username: **admin**
- Password: **admin1pass#**

Or to a basic user account with:

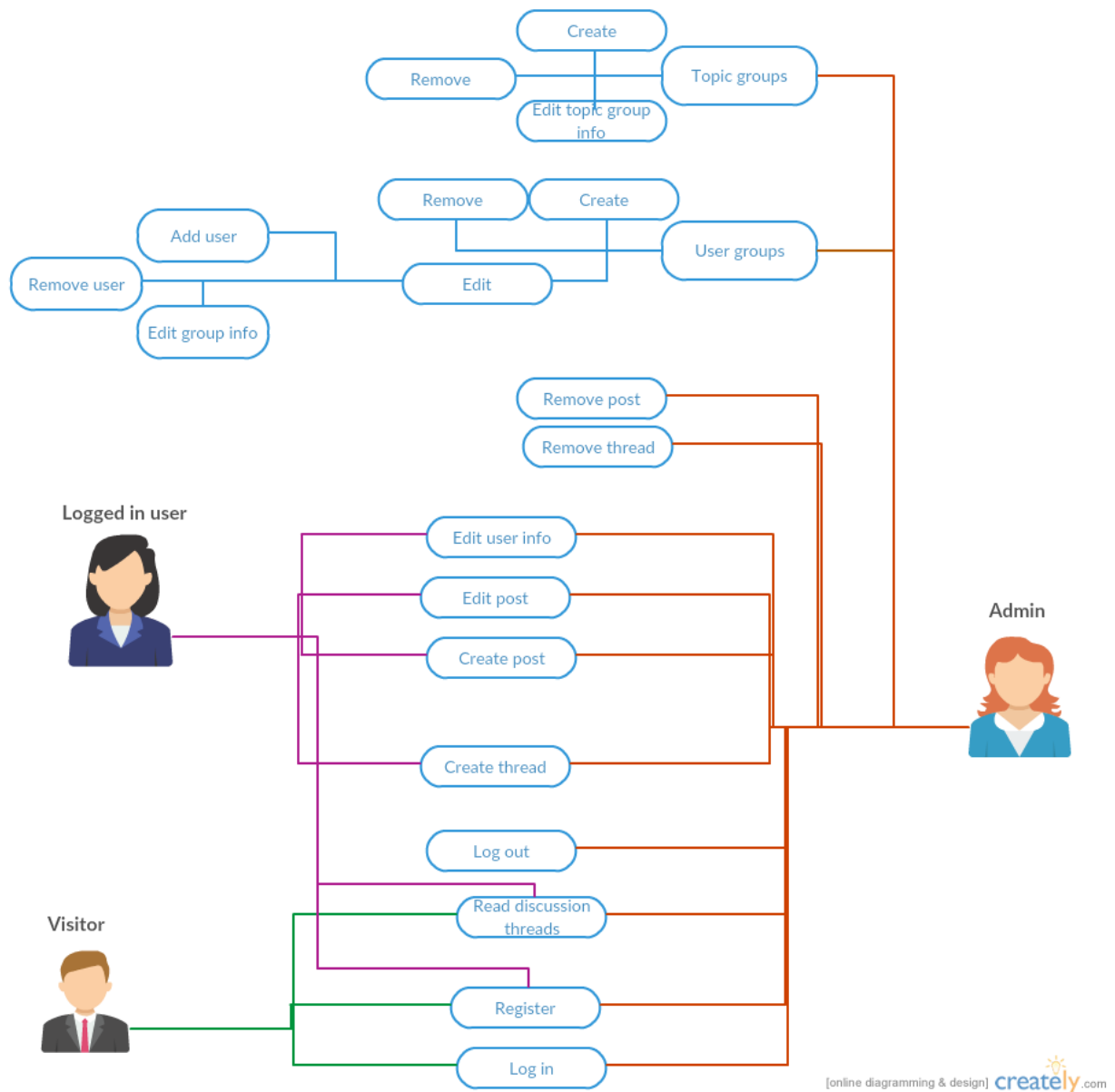
- Username: **user1**
- Password: **user1pass#**

The visuals are tested in Firefox (version 45.0) and Chrome (version 49.0.2623.108)

Use cases

There are three kind of users:

- Visitor is any user who hasn't logged in (and potentially hasn't registered).
- Logged in users are basic users of the site.
- Admin users are administrators with full rights to everything.



A basic logged in user can only edit their own posts and their own user info. Admin users can edit any post and user's info.

Use case descriptions:

Visitor:

- Read discussion threads
 - Visitors can freely read any threads on the forum
- Other functions:
 - Registering
 - Logging in

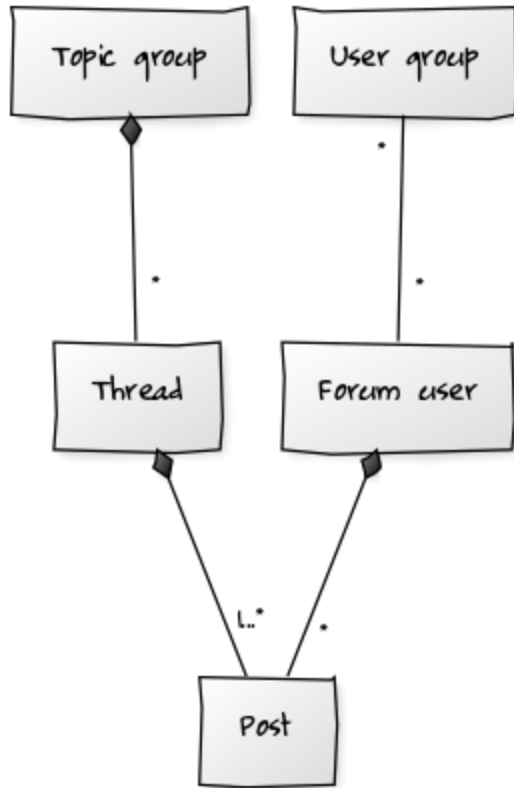
Logged in user:

- Read discussion threads
- Create thread
 - Users can post new threads that anyone can read
 - Thread must contain a post (that will be the first one in the thread)
 - Thread must contain a title
- Create post
 - Belongs to a thread
- Edit post
 - User can freely edit any post of their own
- Other functions:
 - Seeing user info (user's own and other's)
 - Editing user's own info

Admin:

- Edit posts
 - Admins can edit posts made by anyone
- Handle user groups
 - Admins can create groups to which users can be put
 - Group info can be edited, and users can be removed from groups
- Handle topic groups
 - Admins can create groups to which topics by which topics can be organized
 - Topic group info can be edited, and they can be removed
- Other functions:
 - Edit any post
 - Edit any user's info
 - Remove any post
 - Remove any thread

Data schema



Done with yuml.me

Data subject: Topic group

Attribute	Type	Description
Name	String, max 50 characters	
Info	String, max 400 characters	
Created	Date	
Edited	Date	

Topic group can have multiple threads.

Data subject: User group

Attribute	Type	Description
Name	String, max 50 characters	
Info	String, max 400 characters	
Created	Date	
Edited	Date	

User group can have multiple users in it.

Data subject: Thread

Attribute	Type	Description
Title	String, max 50 characters	
Created	Date	
Edited	Date	

Thread can have multiple posts in it, and must contain at least one post.

Data subject: Forum user

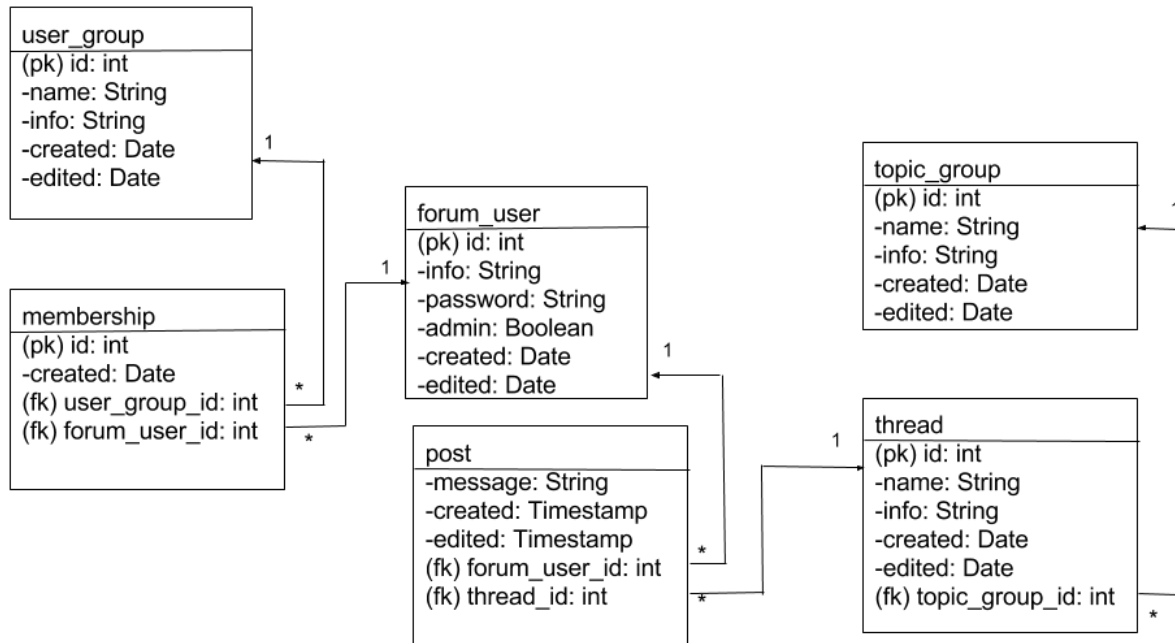
Attribute	Type	Description
Username	String, max 50 characters	
Password	String, max 100 characters	
Info	String, max 400 characters	
Admin	Boolean	
Created	Date	
Edited	Date	

User can have multiple posts and can be in multiple user groups.

Data subject: Post

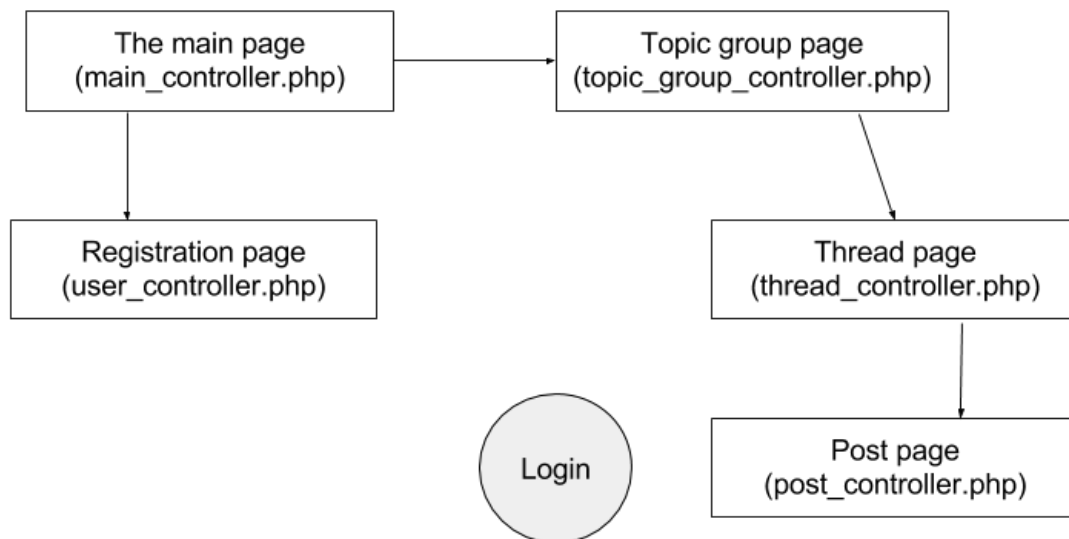
Attribute	Type	Description
Message	String, unlimited length	
Created	Timestamp	
Edited	Timestamp	

Database relation diagram



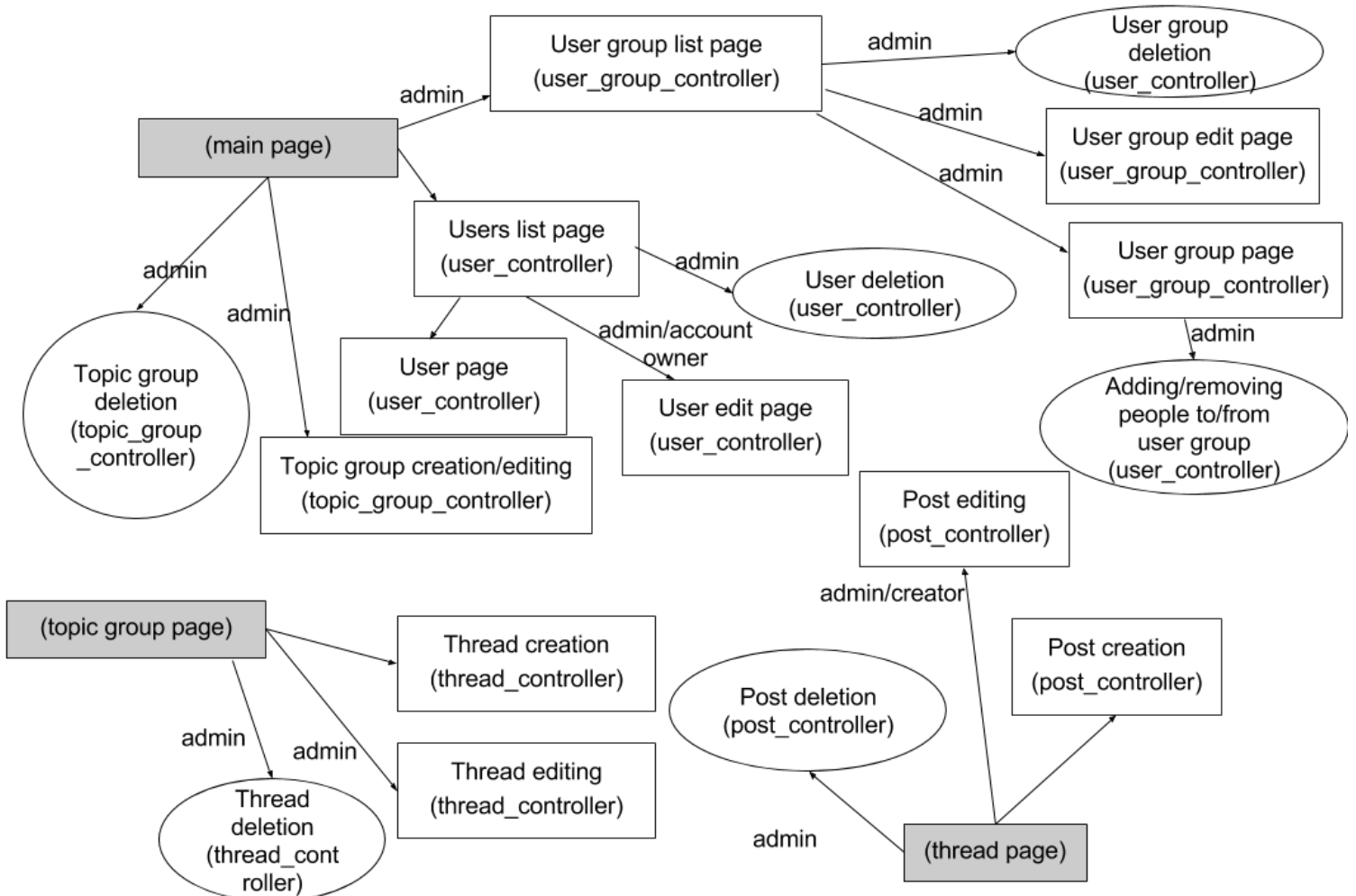
User interface components

Public UI components



Logging in is marked only as a function with no origin place, as it can be done from any page. There is a separate login page also, but it's showed only if the login fails.

You can't see it in the picture, but you can also logout from any page. Each user can also access their user account's info and edit page from any page. Any logged in user can also access any user's post history - links to each post's thread are on the user's profile page.



The general structure of the application

The application follows MVC pattern. Controllers, views and models are located in directories of the same name (for example, controllers are in directory *controllers*). Views are organized in directories each containing the views of the same resource (for example, *views/users* contains views for editing, creating, showing and listing users). Libraries used are located in directory *lib* and in directory *vendor* and settings are in file *settings.php*. Configuration files are in directory *config* and SQL files (including the table creation statements) in directory *sql*. Self created assets, such as images, Javascript and CSS are located in directory *assets*.

All file and directory names are on small case. Different words in file names are separated by an underscore, and different words in directory names are separated by hyphens.

Session related functions (logging in and logging out) are placed in the session controller.

The application uses JQuery to get confirmation from user when they are trying to delete anything.