pastebin-django

Introduction

A pastebin web application programmed in Python using Django web framework. The web application allows both visitors and registered users to upload and view pastes. Registered users can also add comments to pastes, delete their own pastes and mark certain pastes as favorites.

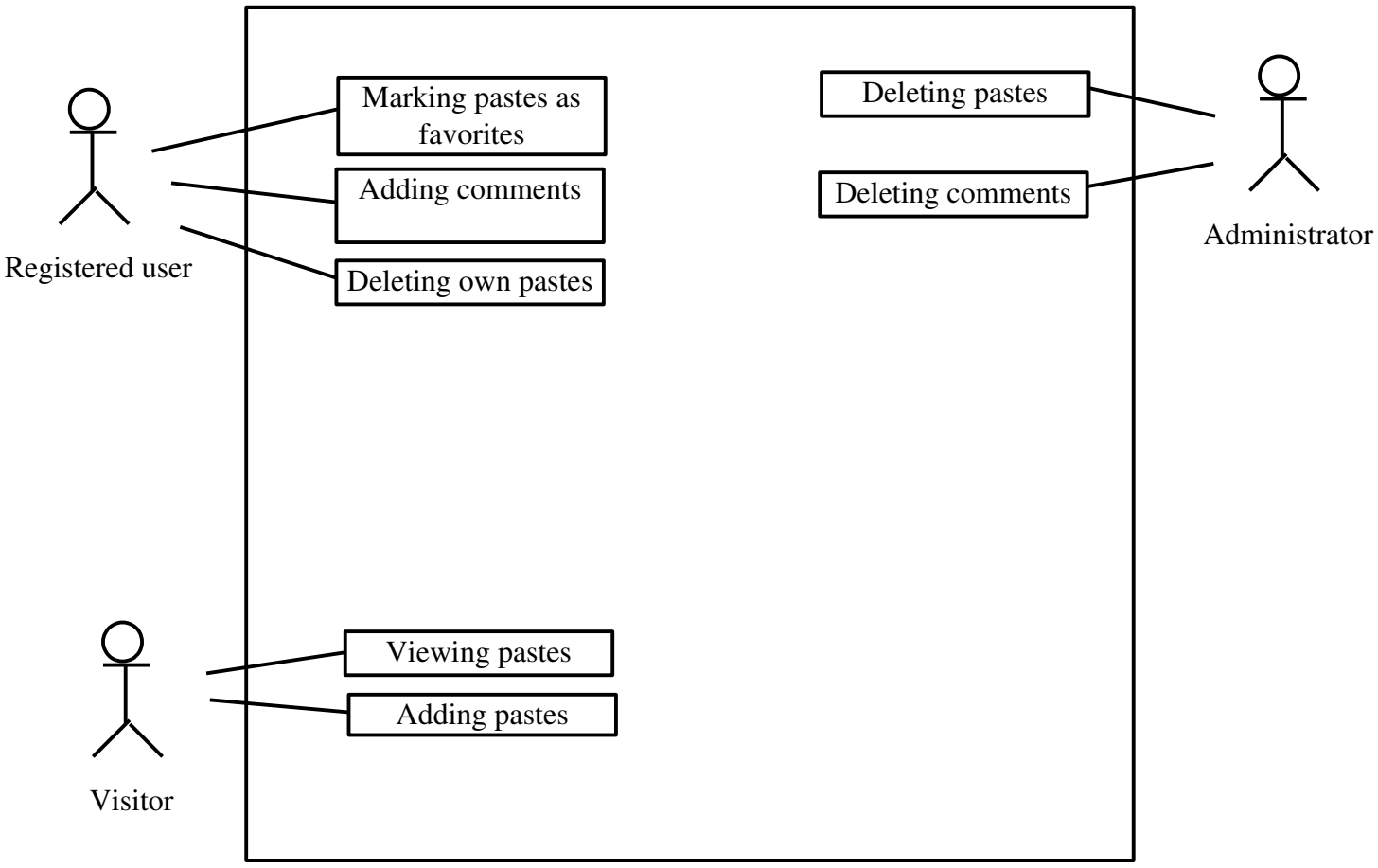
Database

Uses PostgreSQL as the relational database. Data is retrieved, added, updated, etc. using raw SQL queries instead of Django's database-agnostic ORM model.

Other details

Uses Django's in-built authentication system for user registration and login, as this reduces the amount of boilerplate code and is likely more secure than a custom-built solution.

A pastebin web application developed in Python using Django



Uploading a paste

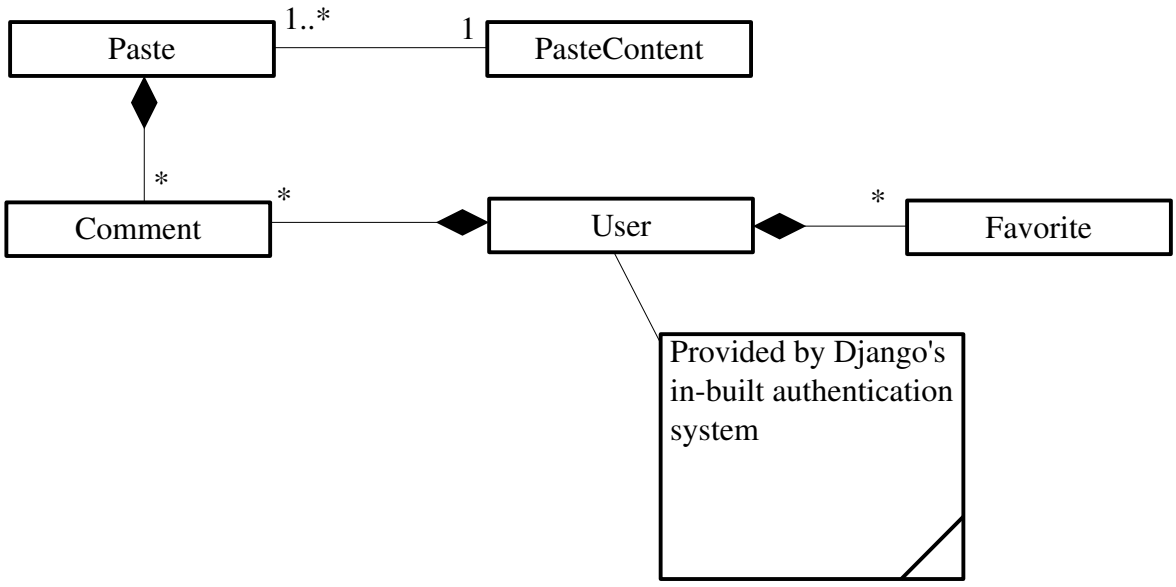
Goal: user uploads a paste which can then be viewed
Users: guest/registered user

1. user fills the form on the front page
2. user submits the form containing the paste's content
    - if a title isn't provided, it is changed automatically to “Untitled”
    - if any text isn't provided, user is returned to the form
3. paste is uploaded and the user is redirected to it

Adding a comment

Goal: user adds a comment to a paste
Preconditions: user is logged in
Users: registered user

1. user fills the comment form
2. user submits the form containing the comment
    - if any text isn't provided, user is returned to the form
3. comment is added and can be viewed by others


Paste
1..*
1
PasteContent
*
Comment
*
User
*
Favorite
Provided by Django's in-built authentication system

Paste

A paste submitted by a guest or a registered user. Text is retrieved based on its hash. Paste may be hidden in which case the only way to view it is to visit the URL directly. Paste may also have an expiration date, after which it can't be viewed at all.

Attribute	Type	Description
id	INTEGER	Primary key of the paste
char_id	CHAR(8)	8-character ID that is used in the URL (eg. G45aaBxy)
user_id	INTEGER	Foreign key of the user who uploaded the paste, can be NULL if the paste was uploaded by a guest
title	VARCHAR(128)	Title of the paste, defaults to “Untitled” if nothing is provided by the uploader
hash	CHAR(64)	Hash used to identify the paste, which saves space as pastes with duplicate content don't need to be stored twice
format	VARCHAR(32)	The format of the text. With plain text this value is “text”, which means the text doesn't have syntax highlighting
expiration_date	TIMESTAMP	A timestamp after which the paste is considered expired. Is NULL if the paste doesn't have an expiration date
hidden	BOOLEAN	Is paste hidden. If the paste is hidden, it won't show up in the Recent pastes list and can only be reached by its URL
submitted	TIMESTAMP	When the paste was uploaded

PasteContent

Paste text is stored as an entry based on its hash, meaning duplicate entries don't take more space.

Attribute	Type	Description
id	INTEGER	Primary key of the paste text
hash	CHAR(64)	Hash identifying the paste
text	TEXT	The paste text
format	VARCHAR(32)	The paste text's formatting. Eg. text with PHP syntax highlighting is “php”, text in raw form without formatting is “text”

Comment

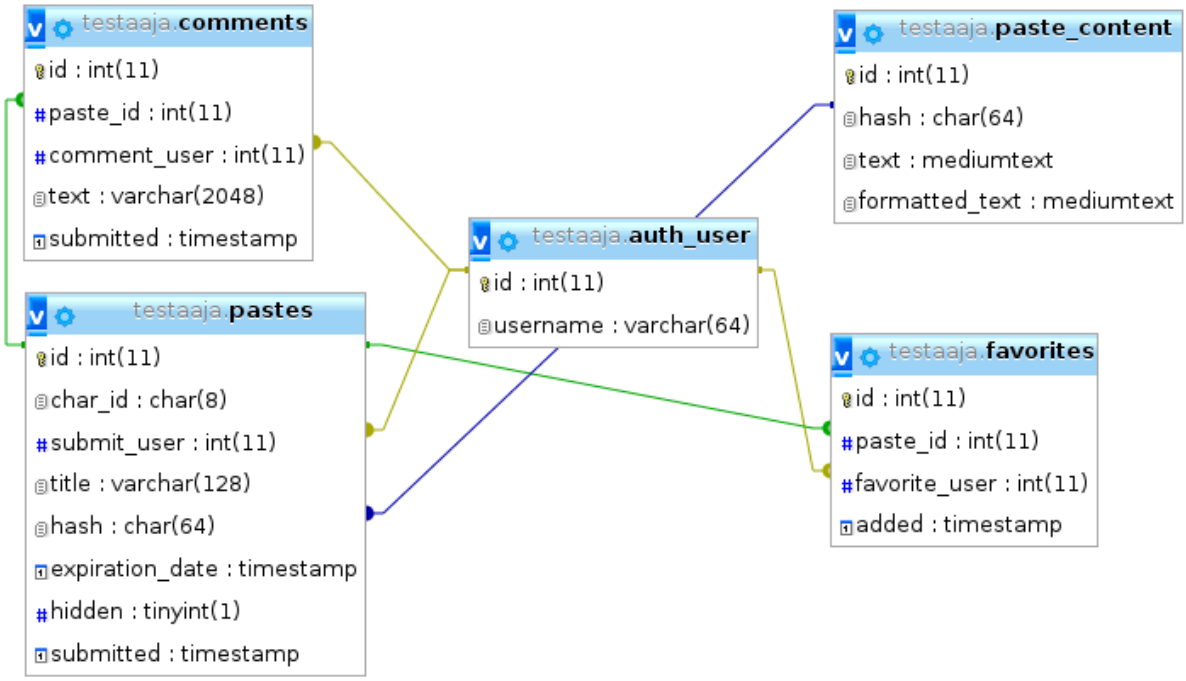
A comment submitted by an user to a certain paste.

Attribute	Type	Description
id	INTEGER	Primary key of the comment
paste_id	INTEGER	Foreign key of the paste this comment was submitted under
user_id	INTEGER	Foreign key of the user who submitted the comment
text	TEXT	The comment text
submitted	TIMESTAMP	When the comment was submitted

Favorite

Registered users can mark pastes as favorites.

Attribute	Type	Description
id	INTEGER	Primary key of the favorite
paste_id	INTEGER	Foreign key of the paste that was added as a favorite
user_id	INTEGER	Foreign key of the user who added the paste as a favorite
added	TIMESTAMP	When the paste was added as a favorite


testaaja.comments
id : int(11)
paste_id : int(11)
comment_user : int(11)
text : varchar(2048)
submitted : timestamp
testaaja.paste_content
id : int(11)
hash : char(64)
text : mediumtext
formatted_text : mediumtext
testaaja.auth_user
id : int(11)
username : varchar(64)
testaaja.pastes
id : int(11)
char_id : char(8)
submit_user : int(11)
title : varchar(128)
hash : char(64)
expiration_date : timestamp
hidden : tinyint(1)
submitted : timestamp
testaaja.favorites
id : int(11)
paste_id : int(11)
favorite_user : int(11)
added : timestamp

Instructions

You can start using the web application immediately by going to http://pastesite.matoking.com. You don't need to register to upload pastes, but deleting pastes you have uploaded requires that you upload the said pastes while logged in. When logged in you can also add any paste into your favorites and add comments to pastes.

You can register easily by clicking the *Register* link on the top-most navbar (it's easy and painless, I promise!) You can login instantly after creating an account.

Once you have logged in, you can view your profile by opening the dropdown menu on the navbar. The pastes you have uploaded will be displayed under *My pastes* while the pastes you have added into your favorites are visible under *My favorites*.

You can delete your own pastes by opening the paste and clicking the *Delete paste* button. You'll have to re-enter your password to confirm deletion.

You can add and remove a paste from your favorites by clicking the *Add to favorites*/*Remove from favorites* button which is visible when viewing the paste and when you are logged in.

You can upload pastes in the site's front page. You can give your paste a title, an expiration date and change its visibility (hidden pastes are visible to other users through its unique URL). You can also add syntax highlighting to the text by selecting the relevant language from the collection of 300+ supported languages.

Structure of the application

The web application was developed following the MVC pattern.

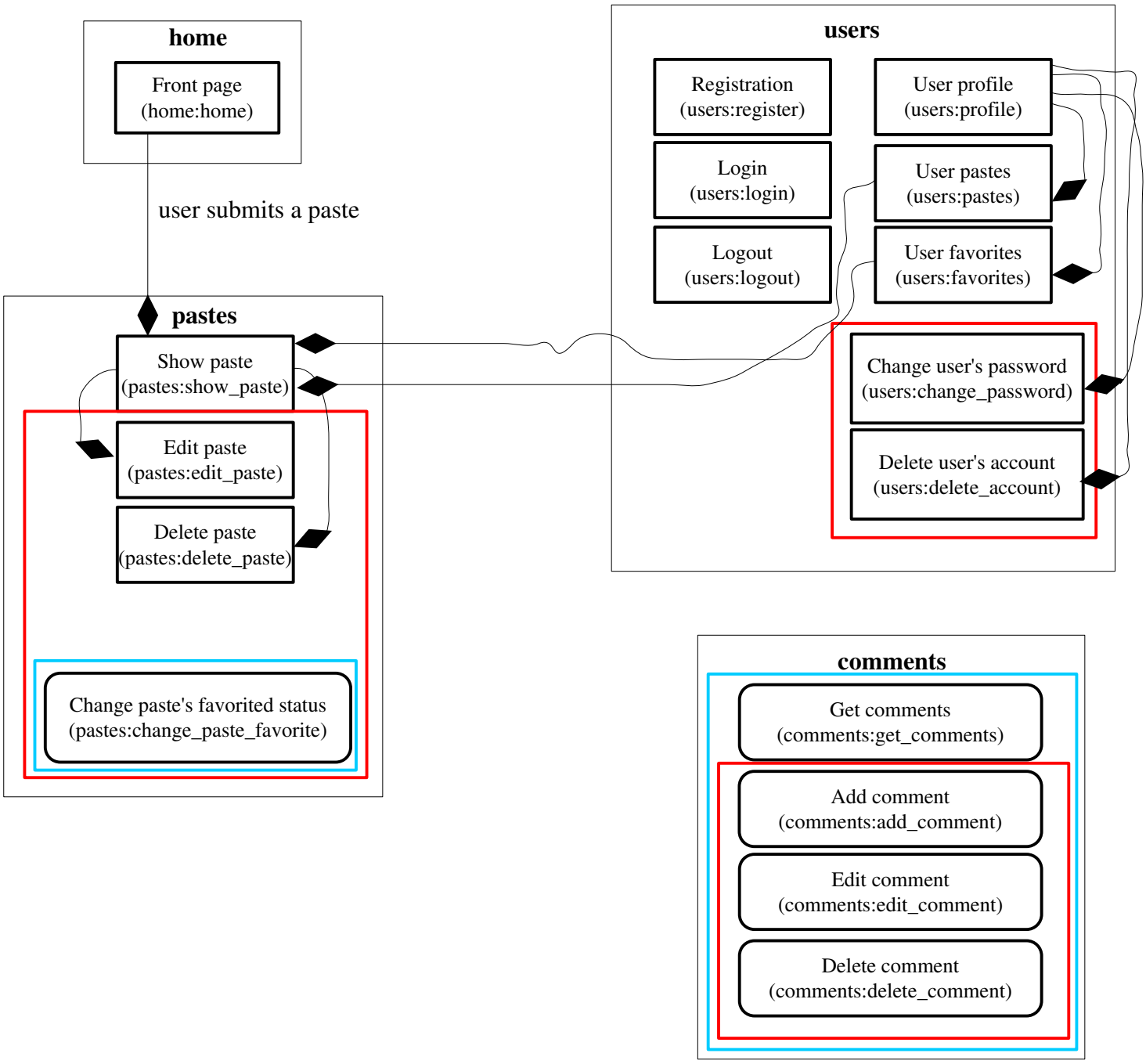
The application has been divided into different apps which handle different parts of the site, such as *pastes*, *comments* and *users,* each of which have their respective views, models and templates. For example, models for the *pastes* app are located in pastes/models.py, views in pastes/views.py and templates in pastes/templates.

The configuration file for the application is located in pastesite/settings.py.

The comment section and the “favorite/unfavorite” button have been implemented using Javascript and *jQuery* library. The scripts can be found in static/js/pastebin-comments.js and static/js/pastebin-favorite.js. The scripts allow the said functions to be done asynchronously without refreshing the page. The scripts send and receive data using normal POST requests where the sent and received data is encoded using JSON.

The web application uses Django's in-built authentication system, which allows user registration and authentication to be done securely more easily than with a custom-built solution. The current user session is stored in the session data.

Python library *Pygments* is used for optional syntax highlighting.



Actions inside the red rectangles require the user to be logged in.

Actions inside the blue rectangles are used by Javascript scripts to retrieve and send information.

The site also has a navigation bar which contains links to every user-related except for account deletion, as well as the front page.