# Simplitt

Documentation

# Introduction

Simplitt is a (very) simplistic Reddit-like PHP-powered webapp.

It has been written by Adriano Ferraguto as part of the “Database” exam of the BS course in Computer Science of the University of Catania, academic year 2015/2016.

Further improvements could be done to the project regardless of the academic obligations.

Simplitt is free software. For more information, read the licence provided in the project repository (GNU GPLv3).

# Requirements analysis

## Problem definition

Simplitt is a web application supporting a Reddit-like website. A user can register to the site. A registered user can submit content (a post) to the site. The homepage must show a list of all posts, sortable by most recent to least recent or by best. Karma is defined as the difference between positive votes (upvotes) and negative votes (downvotes) given by the users of the site to a post. Each post can have comments submitted by the users. The karma of a user is defined by the sum of the karma of all the posts that user made. Each user has a profile page with the karma count and all his posts. A user can delete any post or comment he made.

## Operations

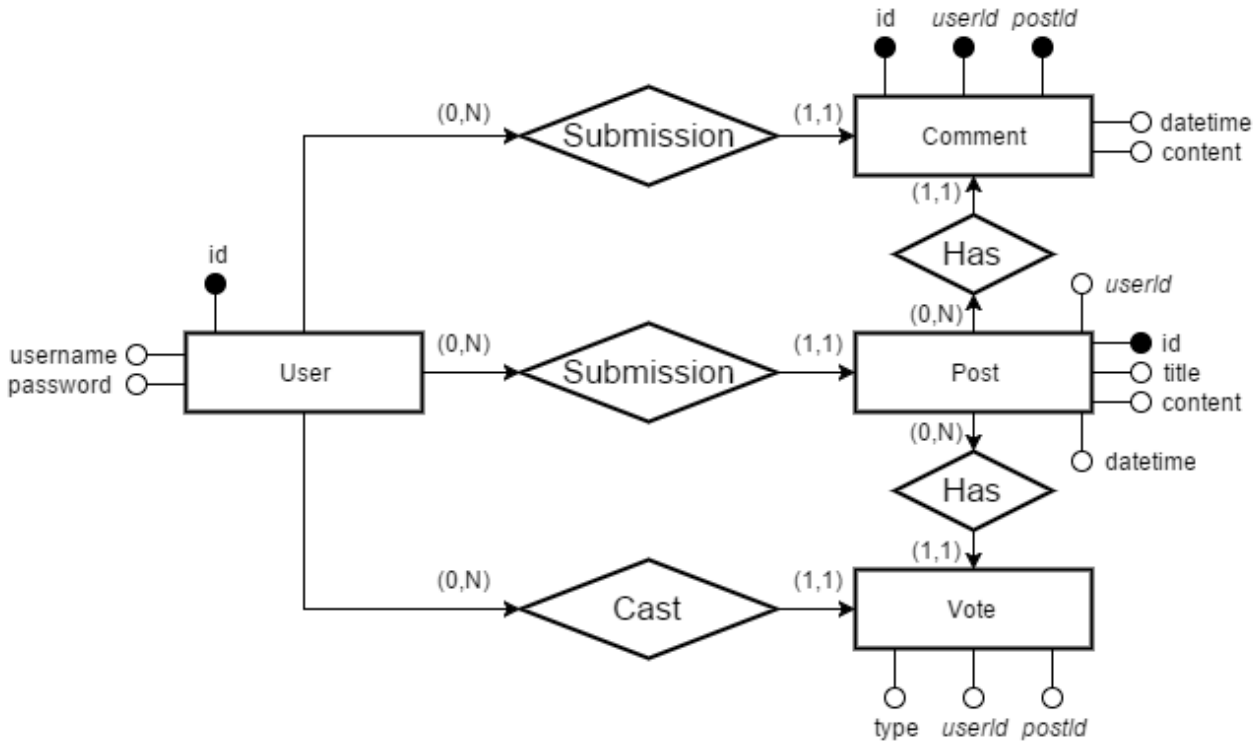
A sample of the most commons operations that the application handles is reported here:

1. A user registers to the website
2. A user submits a post
3. A user votes a post
4. A user changes his vote to a post
5. A user comments a post
6. A user deletes a post

7. A user deletes a comment
8. A user changes his password

# Logical design

## Entity-Relation Schema




## Entities glossary

| Entity | Description | Attributes | Identificator |
|---|---|---|---|
| User | A user of the website | Id, username, password | id |
| Post | A post submitted by a user to the website | Id, userId, title, content, datetime | id |
| Comment | A comment submitted by a user to a post | Id, userId, postId, content, datetime | id |
| Vote | A vote casted by a user to a post | postId, userId, type | (postId, userId) |

## Relations glossary

| Relation | Description | Involved entities |
|---|---|---|
| Post submission | A post submission from a user to the website | User(0,N)<br><br>Post(1,1) |
| Comment submission | A comment submission from a user to a post | User(0,N)<br><br>Comment(1,1) |
| Vote cast | A vote casting from a user to a post | User(0,N)<br><br>Vote(1,1) |
| Post has comment | A comment belonging to a post | Post(0,N)<br><br>Comment(1,1) |
| Post has vote | A vote belonging to a post | Post(0,N)<br><br>Vote(1,1) |

# Physical design

## ‘users’ table

```
CREATE TABLE users
(
   id INT(11) PRIMARY KEY NOT NULL,
   username VARCHAR(25) NOT NULL,
   password VARCHAR(250) NOT NULL
);
CREATE UNIQUE INDEX users_id_uindex ON users (id);
CREATE UNIQUE INDEX users_username_uindex ON users (username);
```

## ‘posts’ table

```
CREATE TABLE posts
(
   id INT(11) PRIMARY KEY NOT NULL,
   userId INT(11) NOT NULL,
   title VARCHAR(100) NOT NULL,
   content VARCHAR(3000) NOT NULL,
   datetime DATETIME NOT NULL,
   CONSTRAINT posts_users_id_fk FOREIGN KEY (userId) REFERENCES
users (id)
);
CREATE UNIQUE INDEX comments_id_uindex ON posts (id);
CREATE INDEX comments_users_id_fk ON posts (userId);
CREATE UNIQUE INDEX posts_id_uindex ON posts (id);
```

## ‘comments’ table

```
CREATE TABLE comments
(
   id INT(11) PRIMARY KEY NOT NULL,
   userId INT(11) NOT NULL,
   content VARCHAR(3000) NOT NULL,
   datetime DATETIME NOT NULL,
   postId INT(11) NOT NULL,
   CONSTRAINT comments_posts_id_fk FOREIGN KEY (postId)
REFERENCES posts (id),
   CONSTRAINT comments_users_id_fk FOREIGN KEY (userId)
REFERENCES users (id)
```

```
);
CREATE INDEX comments_posts_id_fk ON comments (postId);
CREATE INDEX comments_users_id_fk ON comments (userId);
```

## ‘postsvotes’ table

```
CREATE TABLE postsvotes
(
   postId INT(11) NOT NULL,
   userId INT(11) NOT NULL,
   type ENUM('UP', 'DOWN') NOT NULL,
   CONSTRAINT postsvotes_posts_id_fk FOREIGN KEY (postId)
REFERENCES posts (id),
   CONSTRAINT postsvotes_users_id_fk FOREIGN KEY (userId)
REFERENCES users (id)
);
CREATE INDEX postsvotes_posts_id_fk ON postsvotes (postId);
CREATE INDEX postsvotes_users_id_fk ON postsvotes (userId);
```