Sadeg Nabil **Web Application Development** **December 2016**
sc15sn **Coursework 2**
200960399 **Hatebook**

Purpose of website

My website is a social media called Hatebook. It allows the different users to display public messages, comment them and “dislike” them. They also have a profile with different informations such as the name, surname, date of birth and the mood sentence. The users can only communicate publicly.

List of features implemented

The user can register or login. Once he/she is logged in, the user stays logged in thanks to the use of “session”. The person can then modify his/her information, password and upload a picture as a profile picture. It is also possible to create a new post, read the different posts available on the website, comment them dislike them and also report them in case that it is inappropriate. The website also displays the location where the post has been submitted from if the user allows the website to access his/her location. It also displays when a post has been submitted.
The user can also Log out by using the Logout button at the right of the navbar and he/she can also delete his/her account.
There is an admin page as well that can only be accessed by the admin user.

Evaluation of implementation

The admin page is secured using a javascript function and so it is not a very strong security. It would be safer to have a server side security for the admin page.
The geolocation is not always accessible by the application since the user has to accept to give his/her location. So a lot of posts don’t have a geolocation.
The date of birth is stored as a String so the user can input anything. It would have been better to restrict the input to a certain format.

Analysis of web application architecture

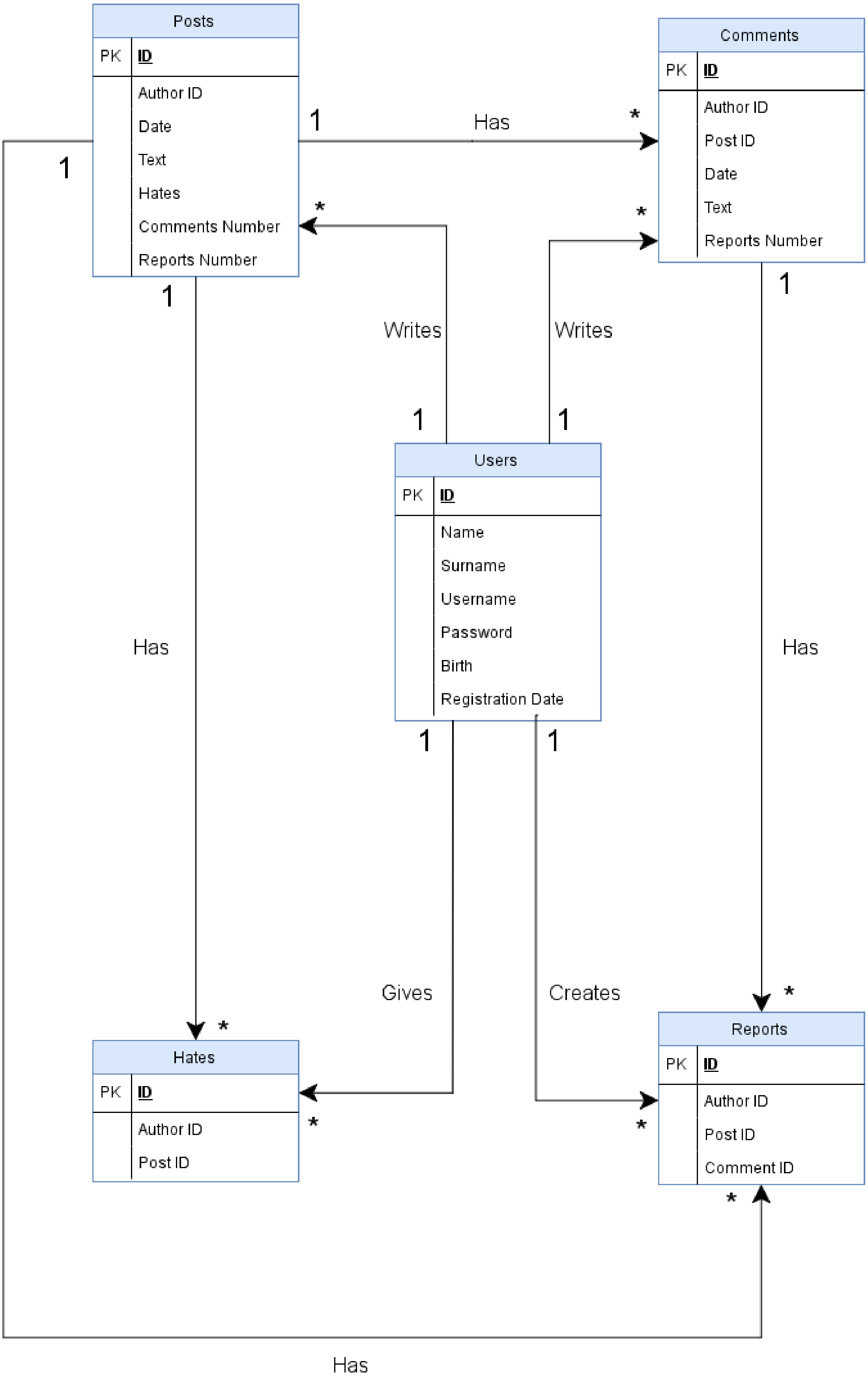
In my application, I have a 3-tier architecture.
The Presentational layer is handled by FlaskWTF for the Forms, Jinja2 for the Templates, CSS3 for the style of the website and Javascript.
The Business logic layer is handled by Flask.route for the Views.
The Data access layer is handled by SQLAlchemy for the Models.
The presentational layer and the data access layer never communicate directly.
The business logic layer passes information to the presentational layer such as the forms, the currently connected user, the differents posts, the different image,...
The business logic layer can access the data in the data access layer with queries.

Security

My Web Applications has potential security issues. First of all, if someone has access to the database containing the different users he/she can retrieve the different passwords. To reduce this issue, the database is always called on the server side. Also the different passwords are converted in hash using hashlib on the client side so the “real” password is never sent to the server and only the hash code is saved in the database.

There is also another potential security issue concerning the profile pictures since a user could upload any type of file. To prevent this, there is a list of supported extensions. If the picture is not using .png or .jpg, the file is not uploaded.

Models



Posts	
PK	**ID**
	Author ID
	Date
	Text
	Hates
	Comments Number
	Reports Number

Comments	
PK	**ID**
	Author ID
	Post ID
	Date
	Text
	Reports Number

Users	
PK	**ID**
	Name
	Surname
	Username
	Password
	Birth
	Registration Date

Hates	
PK	**ID**
	Author ID
	Post ID

Reports	
PK	**ID**
	Author ID
	Post ID
	Comment ID