

University of Cebu-Banilad  
College of Computer Studies  
IT-ELPHP  
Lec -11:30-12:30 PM SAT  
Lab - 1:30-4:30 PM SAT

**Libertas - Blogging Web System**

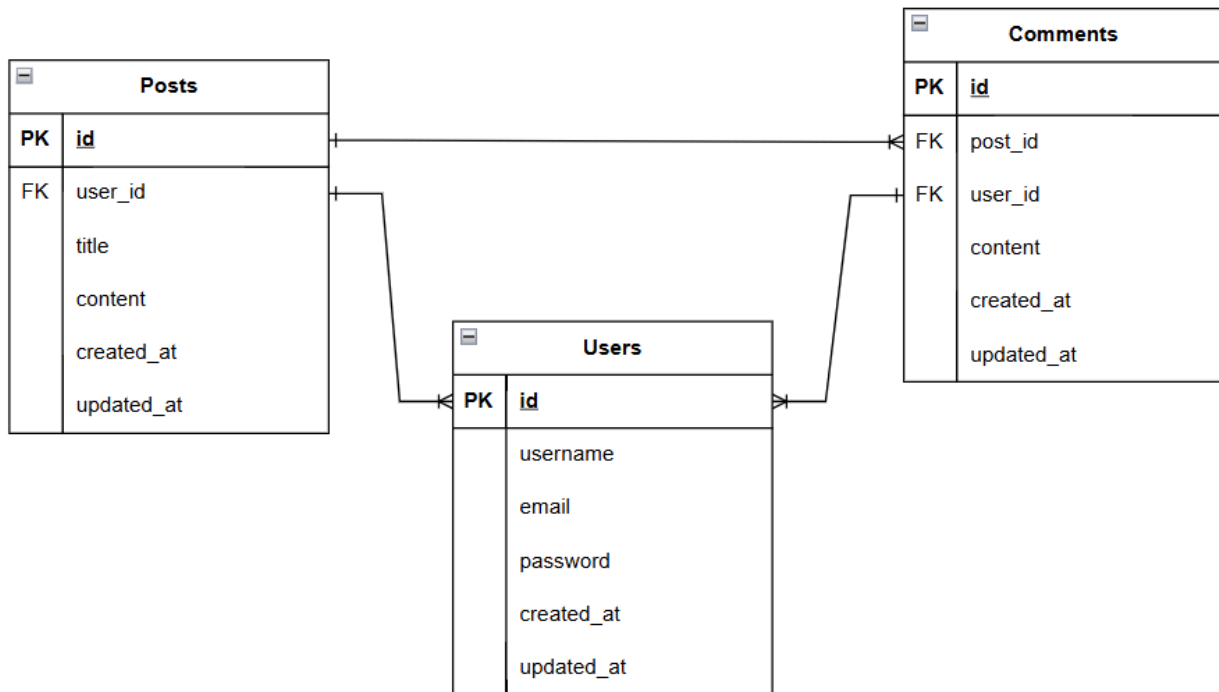
Submitted by:

Gayle Khazie Justo  
Christine Nicole Quiapo  
Francis Ryan Gelira  
Bill John Miraflores  
Marc Jabonite  
Claire Barrientos

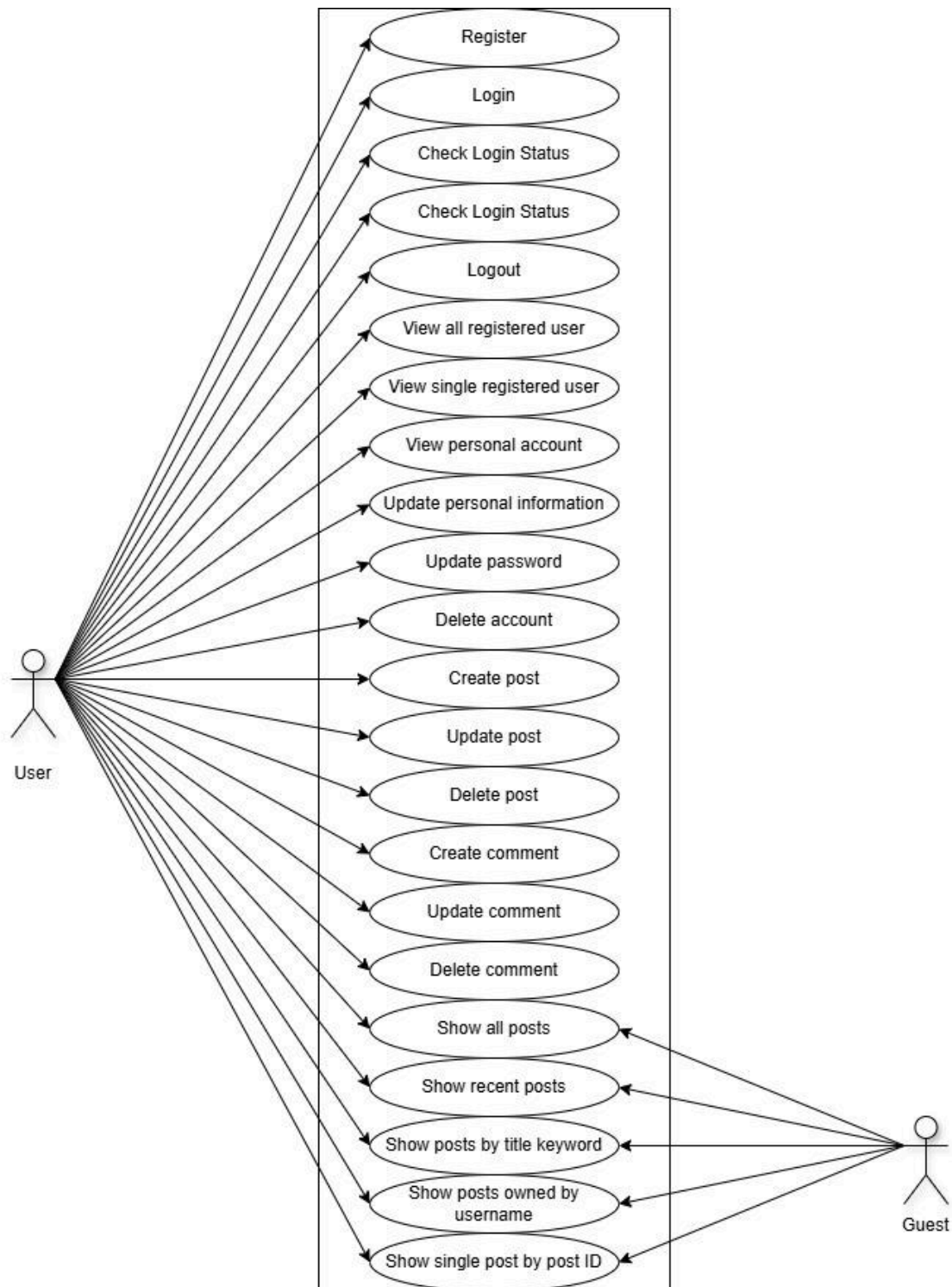
Submitted to:

Mr. Richyll Son

## Entity Relationship Diagram



## Use Case Diagram



## Database Design

Users	
<b>id</b>	<b>integer</b>
username	string
email	string
password	string
created_at	timestamp
updated_at	timestamp

Posts	
<b>id</b>	<b>integer</b>
user_id	integer
title	string
content	text
created_at	timestamp
updated_at	timestamp

Comments	
<b>id</b>	<b>integer</b>
post_id	integer
user_id	string
content	text
created_at	timestamp
updated_at	timestamp

## Users

Column	Type	Constraints
id	integer	Primary Key, Auto-Increment
username	string	Required, Unique
email	string	Required, Unique
password	string	Required (hashed)
created_at	timestamp	Required
updated_at	timestamp	Required

## Posts














Column	Type	Constraints
id	integer	Primary Key, Auto-Increment
user_id	string	Foreign Key → user.id, ON DELETE Cascade
title	string	Required
content	string	Required
created_at	timestamp	Required
updated_at	timestamp	Required










## Comments

Column	Type	Constraints
id	integer	Primary Key, Auto-Increment
post_id	integer	Foreign Key → post.id, ON DELETE Cascade
user_id	integer	Foreign Key → user.id, ON

		DELETE Cascade
content	text	Required
created_at	timestamp	Required
updated_at	timestamp	Required














## QA Testing Results




Test Case ID	Scenario	Method	Endpoint	Request Body	Expected Result	Status	Comments
1	Register with new user	POST	/api/register	Username Email password	Success	 Pass	User successfully registered. No issues encountered.
2	Register with existing data	POST	/api/register	Username Email password	Validation Error	 Pass	Data already exists; appropriate validation error message returned.
3	Login with existing data	POST	/api/login	Email password	Success	 Pass	User logged in successfully, token returned.
4	Login with invalid data	POST	/api/login	Email password	Invalid Credentials	 Pass	Login failed due to invalid credentials, error message returned as expected.
5	Send a token to check login status	POST	/api/status	Authorization: bearer token	Success	 Pass	Successfully received user info based on token.
6	Send invalid/deleted/outdated Token to check login status	POST	/api/status	Authorization: bearer token	Unauthenticated	 Pass	Token is invalid or expired, correctly returns unauthenticated response.
7	Send token to logout	POST	/api/logout	Authorization: bearer token	Success	 Pass	User successfully logged out.
8	Send invalid token to logout	POST	/api/logout	Authorization: bearer token	Unauthenticated	 Pass	Invalid token handled correctly, user remains logged in.
9	Fetch authenticated user info and posts	GET	/api/me	Authorization: bearer token	Success	 Pass	User info and posts fetched successfully.
10	Fetch user info with invalid token	GET	/api/me	Authorization: bearer token	Unauthenticated	 Pass	Invalid token handled properly with unauthenticated response.
11	Update info with valid data and token	PATCH	/api/me	Authorization: bearer token Username email	Success	 Pass	User info updated successfully.
12	Update info with invalid token	PATCH	/api/me	Authorization: bearer token Username email	Unauthenticated	 Pass	Invalid token returned unauthenticated response as expected.
13	Update info with valid token but required fields are not met	PATCH	/api/me	Authorization: bearer token Without	Validation Error	 Pass	Follow required fields

				Username Or email			
14	Update password with valid token and data	PATCH	/api/me/password	Authorization: bearer token Current_password new_password new_password_confirmation	Success	 Pass	Password updated successfully.
15	Update password with invalid token but valid data	PATCH	/api/me/password	Authorization: bearer token Current_password new_password new_password_confirmation	Unauthenticated	 Pass	Invalid token handled correctly, user remains unauthenticated.
16	Update password with valid token but invalid fields	PATCH	/api/me/password	Authorization: bearer token Current_password new_password new_password_confirmation	Validation Error	 Pass	Validation error occurs when password fields are invalid.
17	Update password with valid token, valid fields but incorrect current password or incorrect confirmation password	PATCH	/api/me/password	Authorization: bearer token Current_password new_password new_password_confirmation	Validation Error	 Pass	Correct error message when current password or confirmation password does not match.
18	Delete account with valid token and password	DELETE	/api/me	Authorization: bearer token password	Success	 Pass	Account successfully deleted after token authentication.
19	Delete account with invalid token or invalid password	DELETE	/api/me	Authorization: bearer token password	Invalid Credentials	 Pass	Invalid token or incorrect password results in the proper error message.
20	Fetch all registered user with valid token	GET	/api/users	Authorization: bearer token	Success	 Pass	Successfully fetched the list of all registered users.
21	Fetch user using username with valid token	GET	/api/users/{username}	Authorization: bearer token	Success	 Pass	User profile fetched successfully based on username.
22	Fetch all users with invalid token	GET	/api/users	Authorization: bearer token	Unauthenticated	 Pass	Invalid token results in unauthenticated response.



23	Fetch user using username with invalid token	GET	/api/users/{username}	Authorization: bearer token	Unauthenticated	 Pass	Invalid token leads to unauthenticated response as expected.
24	Create post with valid token and required fields	POST	/api/posts	Authorization: bearer token Title content	Success	 Pass	Post created successfully with correct token and valid fields.
25	Create post with invalid token and invalid required fields	POST	/api/posts	Authorization: bearer token Title or content	Unauthenticated	 Pass	Invalid token results in unauthenticated error.
26	Create post with valid token and invalid required fields	POST	/api/posts	Authorization: bearer token Title or content	Validation Error	 Pass	Validation error occurs when required fields are not provided.
27	Update post with valid token and valid required fields	PATCH	/api/posts/{id}	Authorization: bearer token Title and content or content or title	Success	 Pass	Post updated successfully with valid token and data.
28	Update post with invalid token and valid required fields	PATCH	/api/posts/{id}	Authorization: bearer token Title and content or content or title	Unauthenticated	 Pass	Invalid token results in unauthenticated error.
29	Update post with valid token and invalid required fields	PATCH	/api/posts/{id}	Authorization: bearer token Title and content or content or title	Validation Error	 Pass	Validation error when required fields are missing.
30	Delete post with valid token	DELETE	/api/posts/{id}	Authorization: bearer token	Success	 Pass	Post deleted successfully with valid token.
31	Delete post with invalid token	DELETE	/api/posts/{id}	Authorization: bearer token	Unauthenticated	 Pass	Invalid token prevents post deletion.
32	Delete post with valid token and unauthorized post	DELETE	/api/posts/{id}	Authorization: bearer token	Unauthorized	 Pass	Unauthorized error when attempting to delete other users' post.
32	Delete post with valid token but unknown id	DELETE	/api/posts/{id}	Authorization: bearer token	Not Found	 Pass	Correct "Not Found" response returned for unknown post id.

33	Create comment with valid token and required fields	POST	/api/posts/{post_id}/comments	Authorization: bearer token content	Success	 Pass	Comment created successfully.
34	Create comment with invalid token and valid or invalid required fields	POST	/api/posts/{post_id}/comments	Authorization: bearer token content	Unauthenticated	 Pass	Invalid token results in unauthenticated error.
35	Create comment with valid token but invalid required fields	POST	/api/posts/{post_id}/comments	Authorization: bearer token	Validation Error	 Pass	Validation error occurs when required fields are not met.
36	Create comment with valid token and required fields but unknown post id	POST	/api/posts/{post_id}/comments	Authorization: bearer token content	Not Found	 Pass	"Not Found" returned when post id does not exist.
37	Update own comment with valid token and valid required fields	PATCH	/api/comments/{id}	Authorization: bearer token Post_id content	Success	 Pass	Comment updated successfully.
38	Update comment with invalid token	PATCH	/api/comments/{id}	Authorization: bearer token Post_id content	Unauthenticated	 Pass	Invalid token prevents comment update.
39	Update other user's comment with valid token and valid required fields	PATCH	/api/comments/{id}	Authorization: bearer token Post_id content	Unauthorized	 Pass	Unauthorized error when trying to update another user's comment.
40	Update comment with valid token and valid required fields but comment doesn't belong to the post or comment unknown id	PATCH	/api/comments/{id}	Authorization: bearer token Post_id content	Not Found	 Pass	"Not Found" returned when comment or post id is invalid.
41	Delete own comment with valid token and valid id	DELETE	/api/comments/{id}	Authorization: bearer token	Success	 Pass	Comment deleted successfully.
42	Delete own comment with invalid token and valid id	DELETE	/api/comments/{id}	Authorization: bearer token	Unauthenticated	 Pass	Invalid token results in unauthenticated error.
43	Delete other users' comment with valid token but	DELETE	/api/comments/{id}	Authorization: bearer token	Unauthorized	 Pass	Unauthorized error when attempting to delete other users' comment.
44	Delete comment with valid token and invalid id	DELETE	/api/comments/{id}	Authorization: bearer token	Not Found	 Pass	"Not Found" returned when invalid comment id is provided.
45	Fetch all post	GET	/api/posts		Success	 Pass	

46	Fetch all post using username	GET	/api/posts/{username}		Success	 Pass	
47	Fetch all post using keyword	GET	/api/posts/title/{title}		Success	 Pass	
48	Fetch all post using id	GET	/api/posts/{id}		Success	 Pass	

## Sequence Diagram

