

William Cao, Saad Bhuiyan, Peihua Huang, Ethan Chen (Team SPEW)
SoftDev1 pd2
P #00: Da Art of Storytelling'
2019-10-21

Team: SPEW

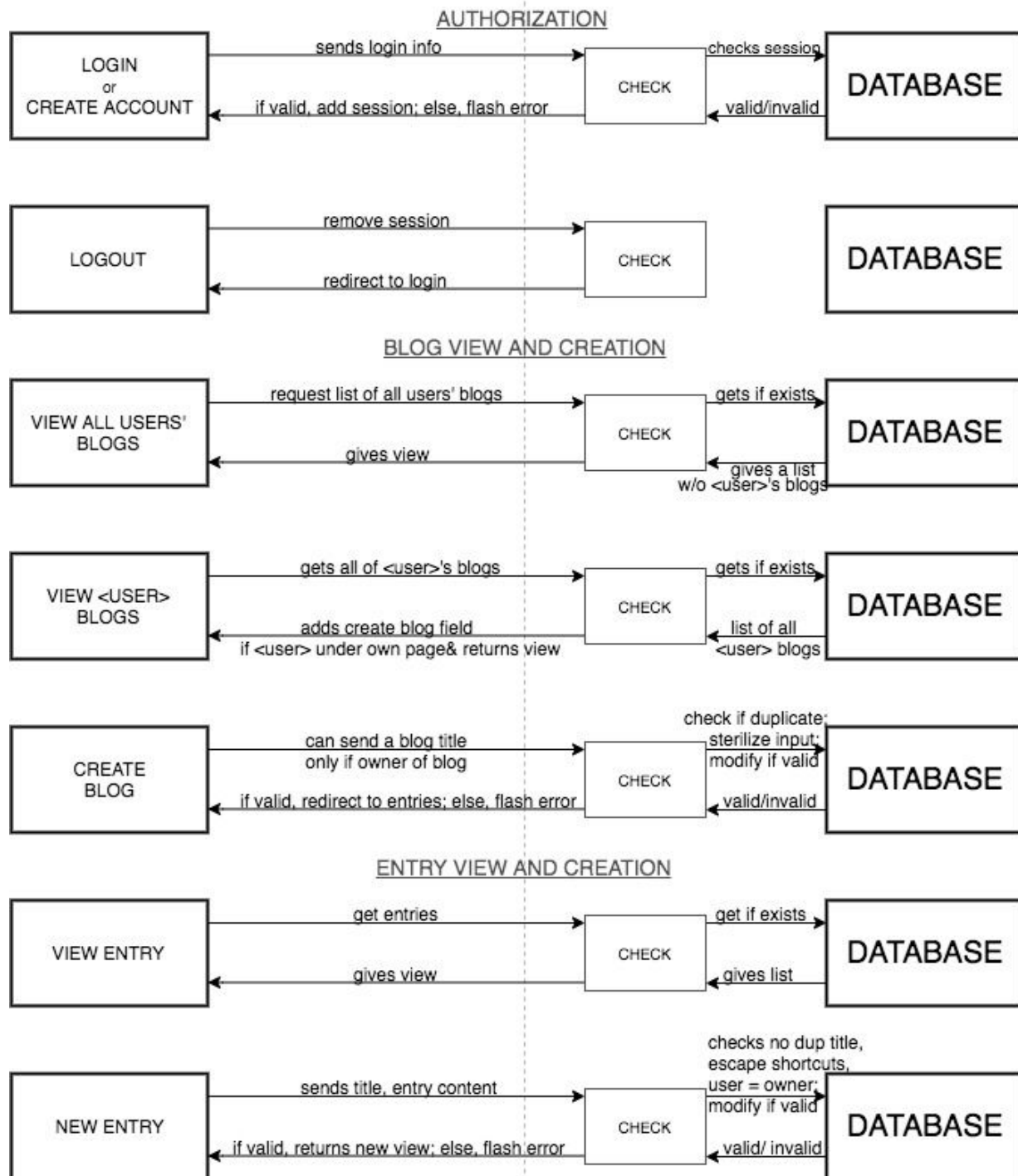
Work division:

Frontend & Testing: William Cao (PM), Ethan Chen
Database: Saad Bhuiyan
Backend: Peihua Huang

Component Relationship

FRONT END (client)

BACKEND (server)

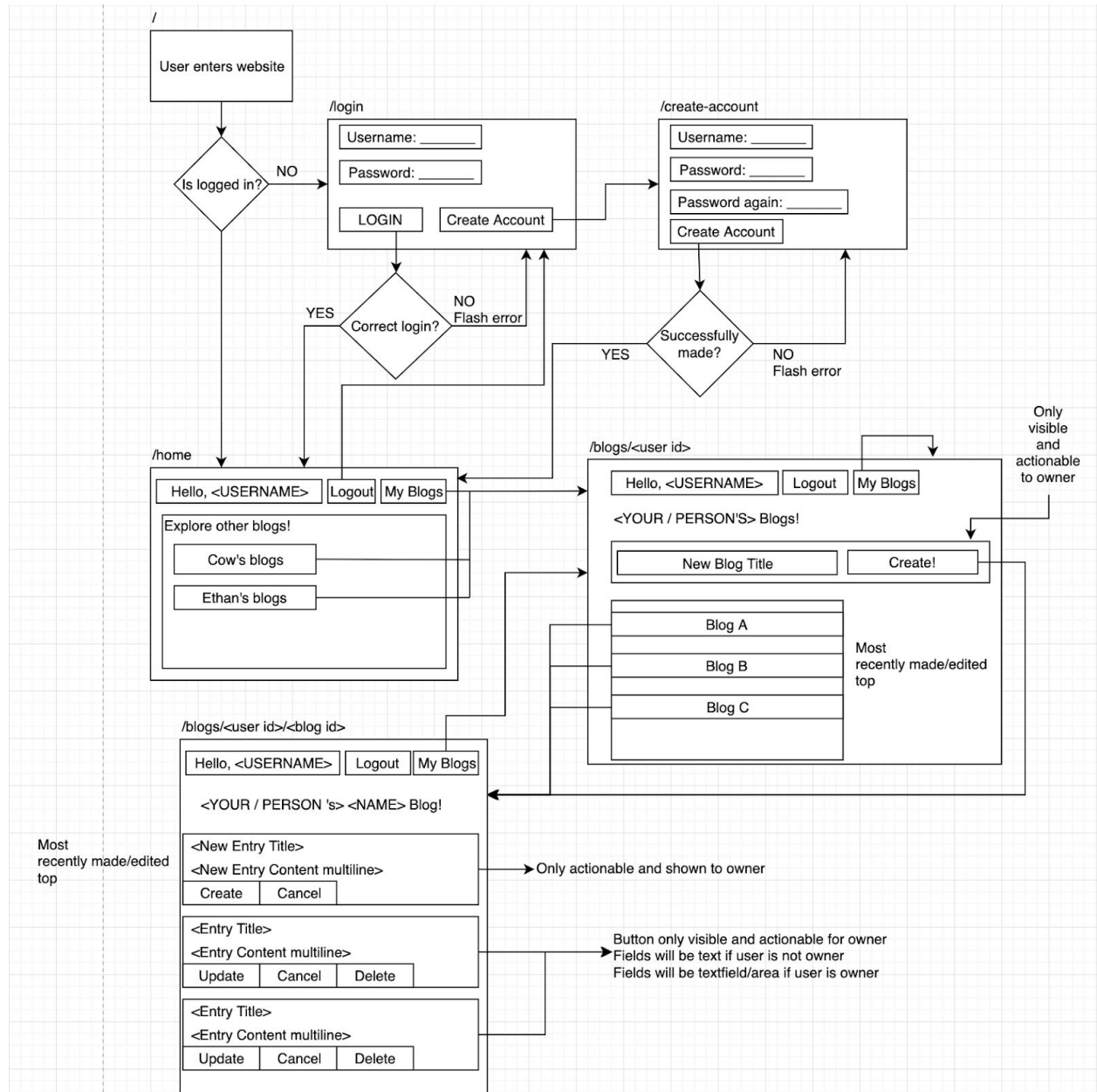


Backend (Flask)

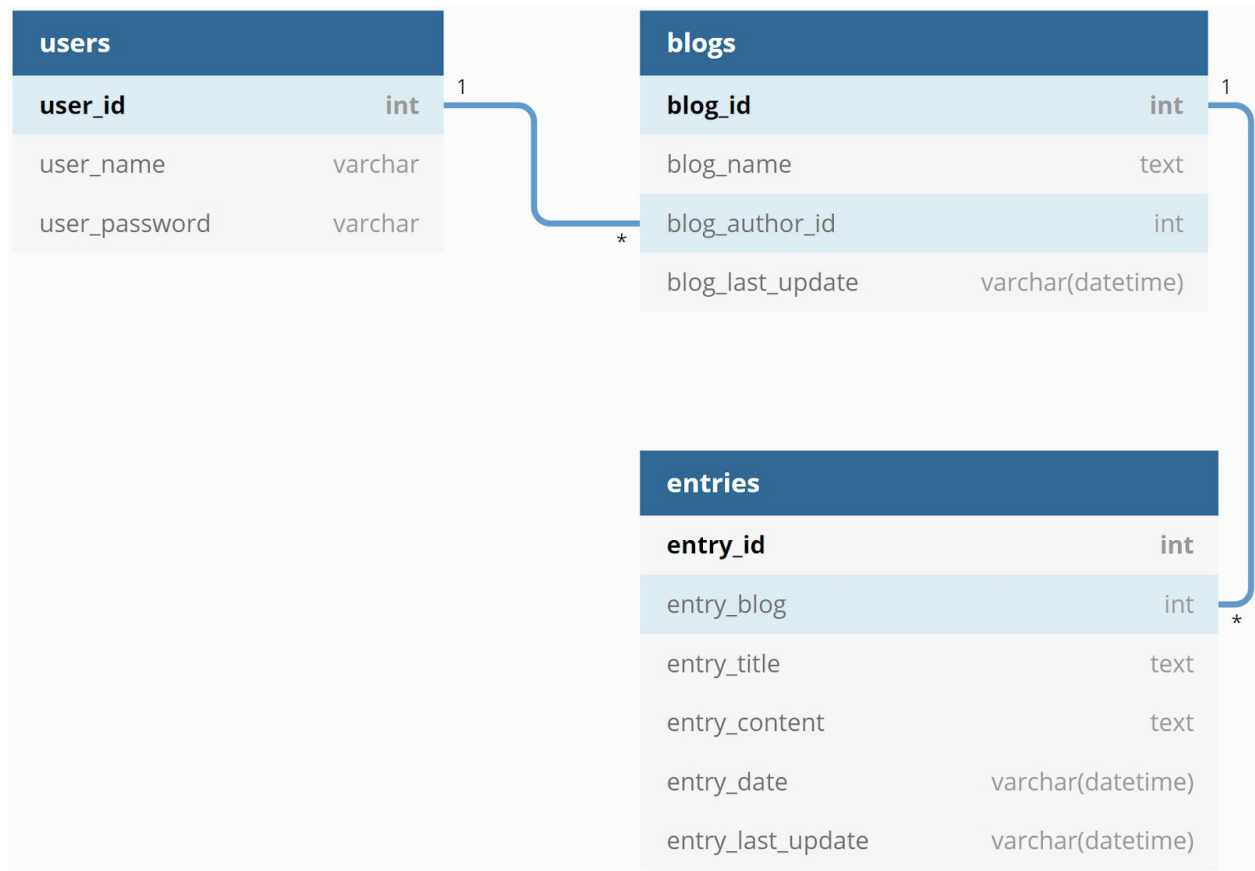
- /
 - Function checks if there is a current session open
 - If there is (meaning user is logged in), redirect to /directory
 - Else, redirect to /login
- /login
 - Load login.html template
 - If username and password is entered, check with database to see if it matches.
 - If it does, add a session (store username and id) and redirect to /directory
 - Else, flash error message
 - If create account button is clicked, redirect to /create-account
- /create-account
 - Load a html template
 - If username not in database, add username, password, and id to database (meaning account was created successfully) and redirect to /blog/<id>
 - Else if bad username (check single quotes, username already exists, passwords don't match), flash error message with the proper error message
- /home
 - Get username from session and display on top of html page
 - Get list of usernames (excluding the logged in user) and display the list
 - If user clicks on the list, get the id of the username clicked and redirect to /blog/<user id>
 - Logout button: if pressed, pop session and go back to /login
 - My Blog button: get user's id from session and redirect user to /blog/<user id>
- /blog/<user id>
 - Loads html template with all the blog titles of the user with that id
 - Check with session to see if user logged in matches the owner of the blogs.
 - If yes, show create! Button and textbox for user to create a new blog
 - If blog title is not found in the database, add the new blog to the database, give it an id, and redirect to /blog/<user id>/<blog id>
 - Else, if duplicate blog, flash error
 - Logout button: if pressed, pop session and go back to /login
 - My Blog button: if pressed and user is not in their own blog, redirect to user's blog
- /blog/<user id>/<blog id>
 - Get the list of entries for that blog in time order (based on last edit time) from the database and display it in chronological order.
 - Check with session to see if user logged in is the owner of the blog
 - If yes, show create, edit, update, cancel and delete button and the choice to input a new entry title and entry content.

- Create button: If entry title is not a duplicate, add the entry to database. Display the new entry above the most recent entry.
 - If entry title is a duplicate: flash error.
- Edit button: If pressed, text in the entry all gets put in a text box where user can edit.
 - Update button: If pressed, the entry text in database is replaced with the new text, timestamp is updated, and entry is now displayed at the top.
 - Cancel button: If pressed, the text box disappears and the entry is reverted to the version stored in database.
- Delete button: Entry title and entry is deleted from the database and thus removed from website.
- Logout button: if pressed, pop session and go back to /login
- My Blog button: if pressed and user is not in their own blog, redirect to user's blog

Front end:



Backend (Database)



Three tables in the database: `users`, `blogs`, `entries` with relevant information in each, as shown in the schema above.

One Python file will contain all the functions that interact with the database. All the other files that need to interact with the database will import the file. Examples of necessary functions include:

- `verifyLogin()` takes login credentials and checks the database `users` table to verify that the login credentials are correct.
- `addLogin()` adds new login credentials to the database `users` table.
- `createBlog()` adds a new blog to the database `blog` table with the `blog_author_id` being the `user_id`.
- `verifyBlogAuthor()` checks if the `user_id` is the same as the `blog_author_id` for a blog. This will be necessary to check if the user can edit the blog.
- `addEntry()` adds a new entry to a blog if the user is the author of the blog.
- `updateEntry()` update the content of an entry.

* Inputs for each function will be decided on as deemed necessary during development.