Team Joe
Jude Rizzo, Derek Leung, Calvin Chu, Jason Zheng

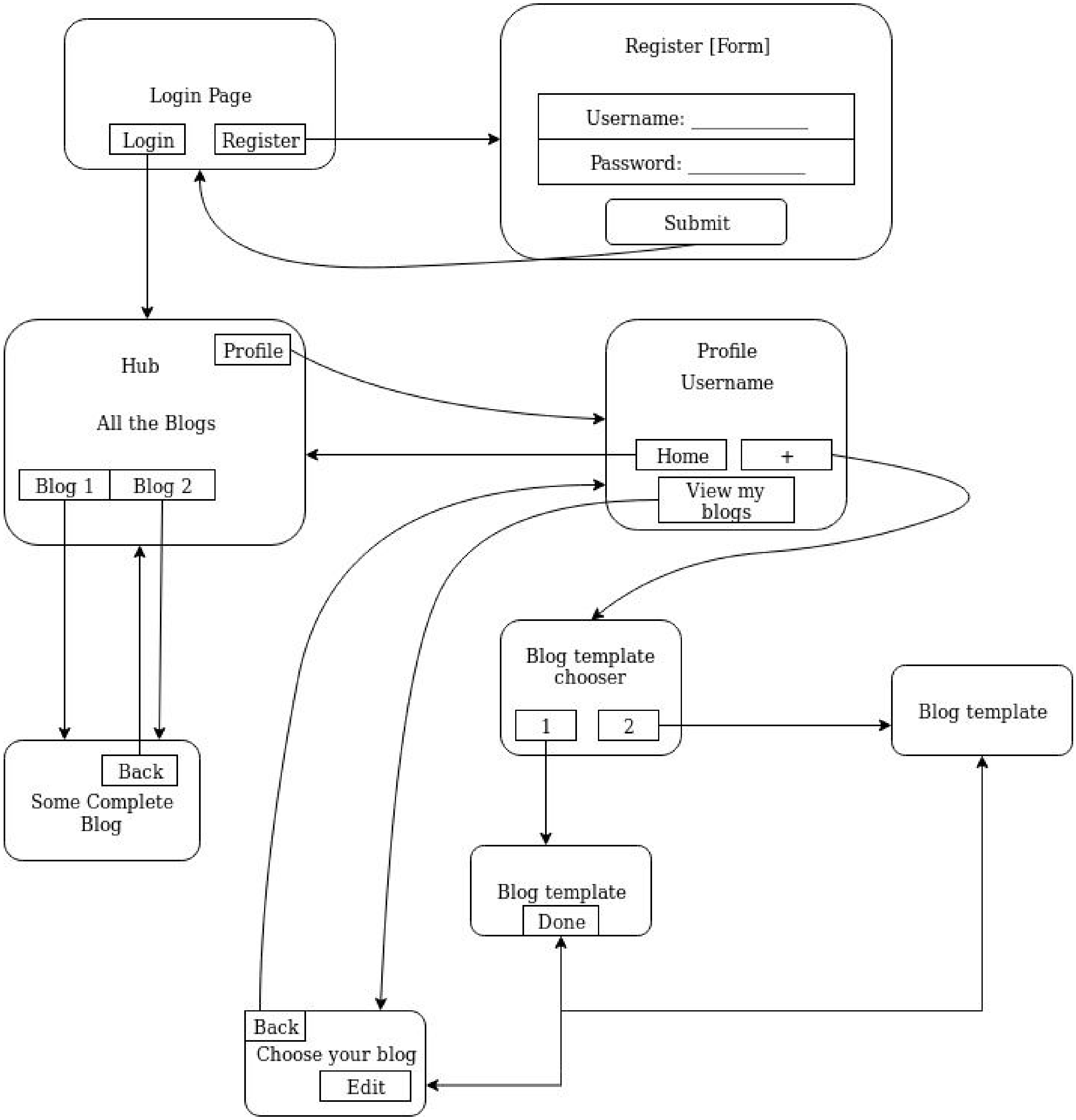
Joe’s Blog

- First there’s a page with register/login button
  - Register button redirects to register page, which redirects back to original page after account is created. Also flashes account created message and adds submissions into the database.
    - If account exists, it will stay on the register page and flash an error message.
  - Login button redirects to login page. If user and pass are wrong it redirects back to the login page. If it’s correct it redirects to main hub page.
- Main hub page has a profile button and links to other blogs
  - Profile button brings to profile
  - Clicking on a blog link brings you to the blog (duh)
- Profile page has 3 buttons
  - One brings you back to main hub
  - One to create a new blog
  - One to view all of user’s blogs
- If a new blog is created, you can choose either of 2 templates. After creating it, you will be sent to the same page as the Profile page button that lets you view your blogs. Completing a blog post will add its title and contents to the Blog Table in the database, as well as an assigned ID number based on the number of blog posts that user has made.

What We’ll Need:

- Flask app with a minimum of 7 routes:
  - Login Page, Register Page, Main Hub, Profile Page, Blog Page, “My Blogs” Viewing Page, Blog Creation Page, etc.
- HTML/Jinja templating for front end
- Database with 2 tables
  - Login Table storing Usernames and Passwords
  - Blogs Table storing Usernames, Blog Topics, and Blog Posts
- Python program to facilitate adding to database tables

SiteMap


Login Page
Login
Register
Register [Form]
Username: ___________
Password: ___________
Submit
Hub
Profile
All the Blogs
Blog 1
Blog 2
Profile
Username
Home
+
View my blogs
Back
Some Complete Blog
Blog template chooser
1
2
Blog template
Blog template
Done
Back
Choose your blog
Edit

Database Organization

- We’d have two databases, one to store username and password, and the other to store who owns which blog as well as the title of the blog and the contents it contains.

Example:

Username	**Password**
Joe	Mama
Derek	Leung

User	**Blog ID**	**Title**	**Contents**
Joe	0	“Something”	“Who’s joe?”
Joe	1	“Something2”	“Joe mama lol”
Derek	2	“Something3”	“Hello”

Minimum Viable Product

- Login Page with login and register button
- Register Page with a submit button
- Page with all the blogs with create blog and view own blog button
- Page to create a new blog
- Page to view own blogs with edit button

EXTRA

- Templates for a new blog
- Cookies
- Search Bar
- Change credentials / delete account

Roles

- **Flask App Creator (Jude and Jason)**
    - Person will create a Python flask application that will have each of the routes we need (mentioned above)
    - Person will make sure that the pages that require you to be logged in cannot be accessed by just typing the extension into the URL if you have not logged in
    - Person will make sure to flash messages when appropriate to do so (i.e. flashing a login error when submitting a wrong username or password on the login page)
- **HTML/Jinja Template Creator (Calvin)**
    - Person will create the HTML templates for each of the routes
    - Person will use forms when appropriate (i.e. login page => username and password inputs, submit button)
    - Person will use Jinja to take in and display variables when necessary (i.e. for the blog post template, use Jinja to display a blog post pulled from the database table)
    - Person will pay attention to aesthetic details
- **Database Creator (Derek)**
    - Person will write a Python application to create appropriate databases and tables, making sure to only add them if they don’t already exist
    - Person will facilitate adding to, editing contents in, and potentially removing from tables