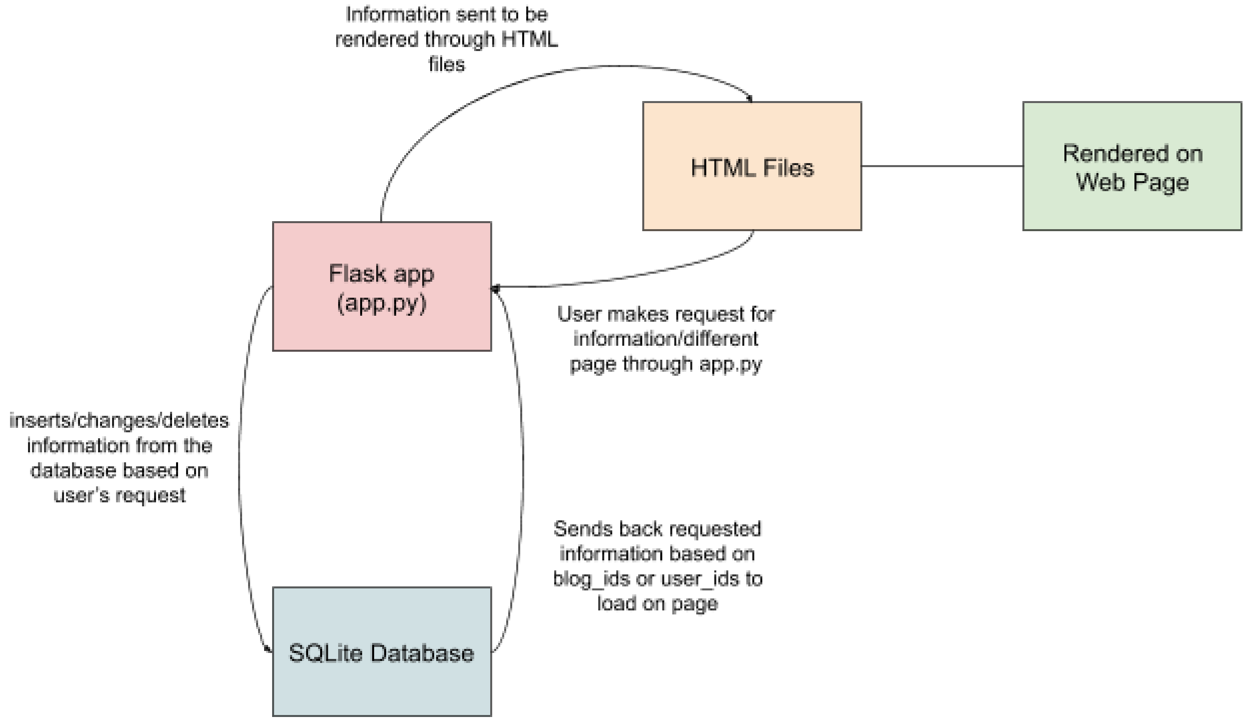
Team SeamLess (Maddy Andersen, Amelia Chin, Erin Lee, Stella Oh)
SoftDev — Rona Ed.
P1 — DESIGN DOCUMENT
20201-01-xx

Program Components and Role of Each

- **SQLite Database**: holds our data and information so we can access it to display on the site
  - updates with edits to blogs and existing entries
  - Creates new tables for new blogs
- **Flask**: the framework for our web-based app, connecting and managing the routes and functions
- **Python**: works with Flask, directs our functions and renders templates
  - Pulls information from SQLite and sends to Flask app to be displayed on page
- **HTML/CSS**: foundation and styling for our front-end page

COMPONENT MAP


Information sent to be rendered through HTML files
HTML Files
Rendered on Web Page
Flask app (app.py)
User makes request for information/different page through app.py
inserts/changes/deletes information from the database based on user's request
Sends back requested information based on blog_ids or user_ids to load on page
SQLite Database

Databases

BLOGS (list of all blogs)

blog_id	name	description	user_id
integer	text	text	integer

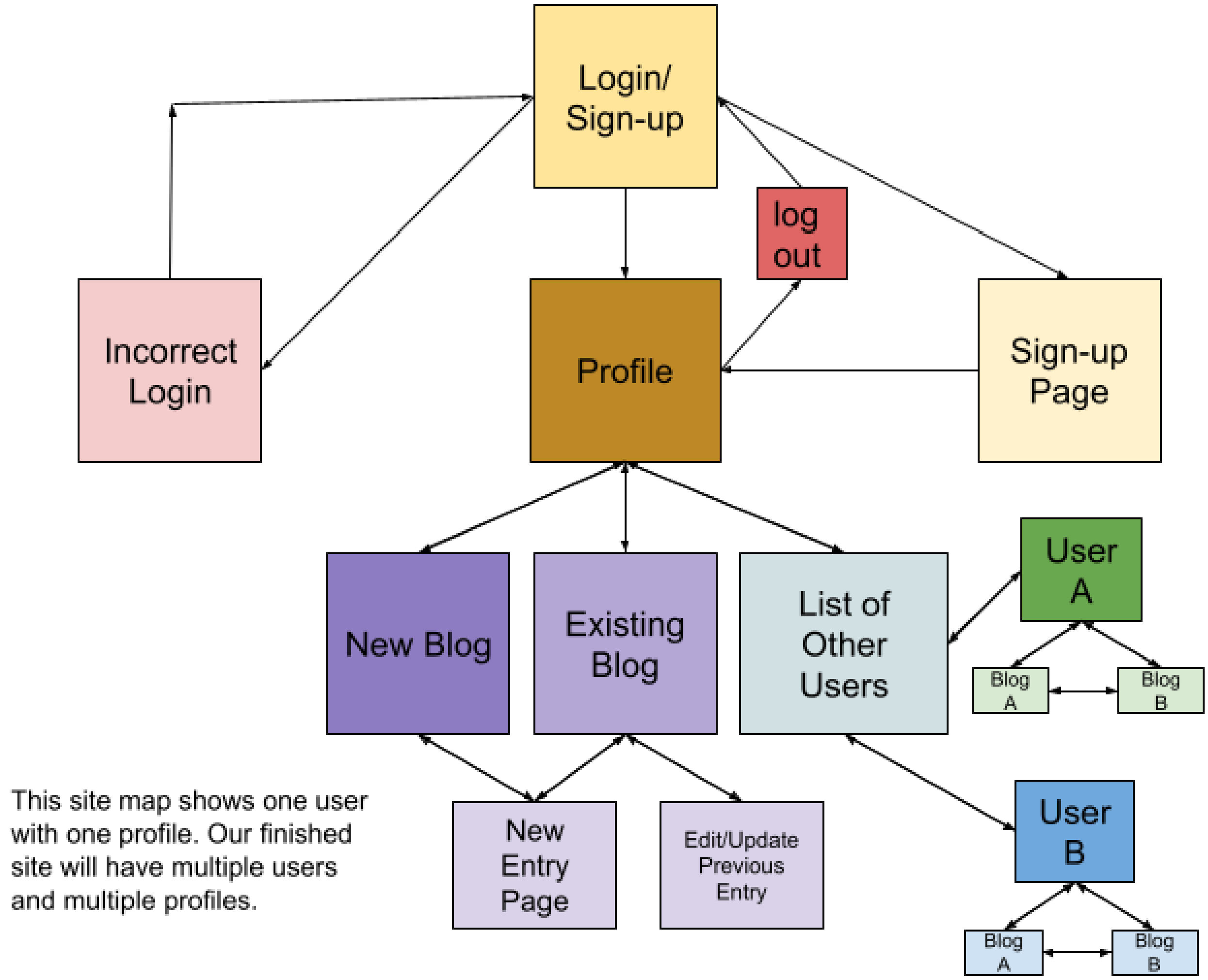
ENTRIES (one table per blog)

entry_id	blog_id	title	content
integer	integer	text	text

USERS (list of all users)

user_id	username	first_name	last_name	password
integer	text	text	text	text

SITE MAP



CSS and Bootstrap Elements

- **fonts**
  - body: Lato
  - h1: Lobster Two, bold
- **colors**
  - background, footer text: #BCD6B4 (light green)
  - footer background, main text: #005C59 (teal)
  - buttons: #005C59 (teal), #67839E (greyish blue), #9E0E1D (fire engine red)
  - button text: #FFFFFF (white)
- **Bootstrap elements**
  - forms with floating labels
  - mb-3 class instead of breaks

    - btn class for all our buttons
    - tables with striped and hoverable rows
    - container class around all content except footers
- **external CSS stylesheet elements**
    - footer class styles footer containing our team name and roster
    - input class gives all forms width of 50%
    - table class gives all tables width of 100%
    - top-btn, btn, red classes color different buttons
    - btn-set class aligns buttons next to each other

Task Breakdown

P0 (Lach On: Maddy Andersen, Dean Carey, Kelly Huang, Erin Lee)

- In general, we all need to figure out how to connect databases to pages (how to access/print tables onto the Flask app (can only view in terminal currently))
- We also need to figure out how to create and manage sqlite tables without csv files
- Everyone will be responsible for creating the python, sqlite, and html files associated with their respective tasks
- **Login/Sign up** (*Dean*) 1/04-05
    - Registering users (setting up database of users and passwords, matched with user's profile link)
    - Adding a row to table of users, passwords, and user information
    - Set up login/register page, sign up page, and incorrect login page
    - Check login info with Sqlite table of users and passwords
    - Link login/sign up page with incorrect login page
    - Link login/sign up page with sign-up page
    - Link login/sign up page with profile
    - Link profile with logout page, link logout page with login/sign up page
    - Add session to keep track of the logged in user
- **Create a new blog, view an existing blog, update/edit blog** (*Maddy and Erin*) 1/05-06
    - Create new blog page
        - Create table
        - Add row to table of blogs
    - View existing blog(s)
        - For each blog, show what is currently in sqlite table (List all entries of a blog's table)
        - Link new entry page
        - Link edit an existing entry page
    - New entry page
        - Add an entry → add a row to the Sqlite table
    - Edit an existing entry page
        - Edit data in Sqlite table
    - Link profile page with new blog page and existing blog(s) page
- **List of all other users (excluding current user) + profile page** (*Kelly*) 1/06-07
    - Create viewable profile page for each user

                - Links to all of user’s blogs
        - List of all users (with links to their viewable profile)
        - Once the current visitor clicks on a profile, they should see a list of user’s blogs
        - Once visitor clicks on a blog, they should be able to read all entries within that blog

P1 (SeamLess: Maddy Andersen, Amelia Chin, Erin Lee, Stella Oh)

- **Bootstrap and CSS for templates**
    - login.html (we worked on this together to get comfortable with Bootstrap and CSS)
    - profile.html, blog.html, masterlist.html (*Erin*)
    - register.html, error.html (*Maddy*)
    - create_blog.html, edit_blog.html (*Stella*)
    - create_entry.html, edit_entry.html (*Amelia*)

Reflection

P0 (Lach On: Maddy Andersen, Dean Carey, Kelly Huang, Erin Lee)

- We ended up working collaboratively for most of the project. While we would break out into groups for small assignments, we did not stick to our assigned roles for the most part.
- Had we had more time, we would have liked to add the ability to delete blogs + entries, and we would have wanted to order blogs + entries by creation date. We also may have included CSS.