TiMBer
Software Development
Period 9
Michela Marchini, Bermet Kalmakova, Tina Chen
Project Manager: Michela Marchini
TiMBer_chenT_marchiniM_kalmakovaB

Cloud Diary

The cloud diary will allow users to keep a user-friendly online diary. Once logged in, users will be directed to a calendar. The calendar will show events the user has logged. By clicking on a specific day users will either be able to add an entry (including pictures and music) for that day or view that day's entry. Entries can also be updated by clicking an edit button on the entry page and submitting new material for that day (you cannot delete an entry you have submitted).

STRETCH GOALS: Users can customize their diary by changing background/text color or font.

Components

Back End

- app.py
    - Sqlite database functions
        - Creates the databases for users and entries
        - Includes functions on adding and retrieving users and entries
        - FUNCTIONS
            - adduser(username, password): returns true if added, false otherwise
            - user_exists(username): returns true if user exists, false otherwise
            - get_password(username): returns password of given user or none if password/user doesn't exist
            - addentry(date, type, data): adds entry based on date; returns true if added, false otherwise;
            - getentries(username, date): returns list of entries listed for that date
            - getevents(username, date): returns a list of events for that date
    - Flask app
        - logged_in()
        - /logout
        - /login
            - user_exists(username)
            - check_pass(username, password)
        - /create
            - adduser(username, password)
        - /home

            - Render calendar.html for user
            - When you click on a calendar space, if empty redirects to input, otherwise redirects to entry for that day
        - /input
            - Renders input.html
        - /update
            - Updates database using inputted data from input.html
        - /entry
            - Renders entry.html with all appropriate data
- APIs
    - Youtube api to play music

Front End

- frame.html
    - Contains frame for all pages including headers
- calendar.html
    - Runs the calendar.js
    - Each box has either a “create an entry”, “view an entry”, or “edit an entry”
- calendar.js
    - Contains boxes for each date as svg elements
- input.html
    - Contains a form to create your entry
    - Elements include: adding photos, captions, choosing a design, choosing a song to play in the background
- entry.html
    - Displays the entry of a specific date based on the form filled out
- login.html
    - Fields to enter username and password to log into an existing account
- create.html
    - Fields to enter username and password to create a new account

Database Schema

Users

TEXT PRIMARY KEY username	**TEXT password**
michelamar	itsmichela
bermet	123456789

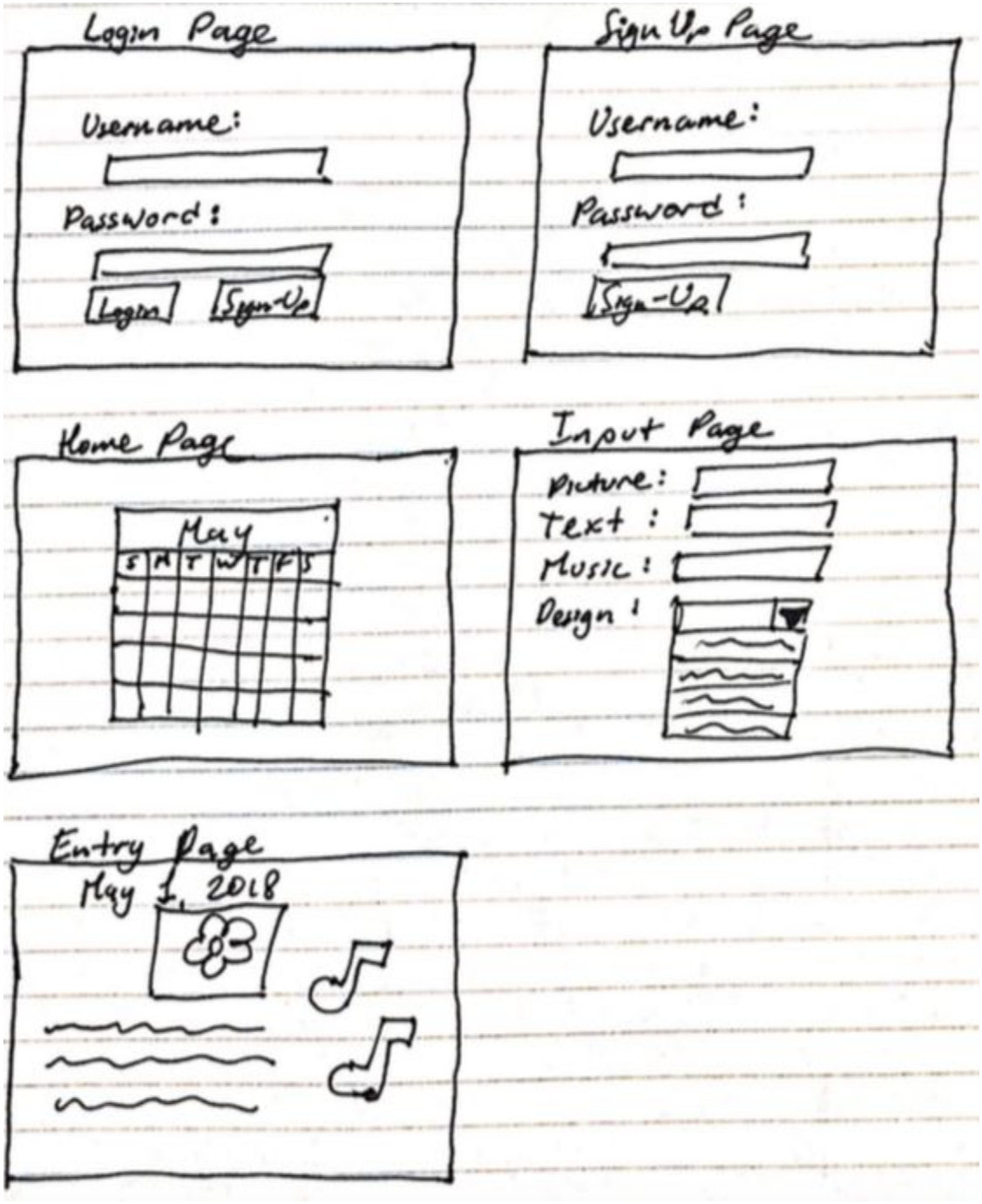
Entry

TEXT username	**NUMERIC date**	**NUMERIC type**	**TEXT data**

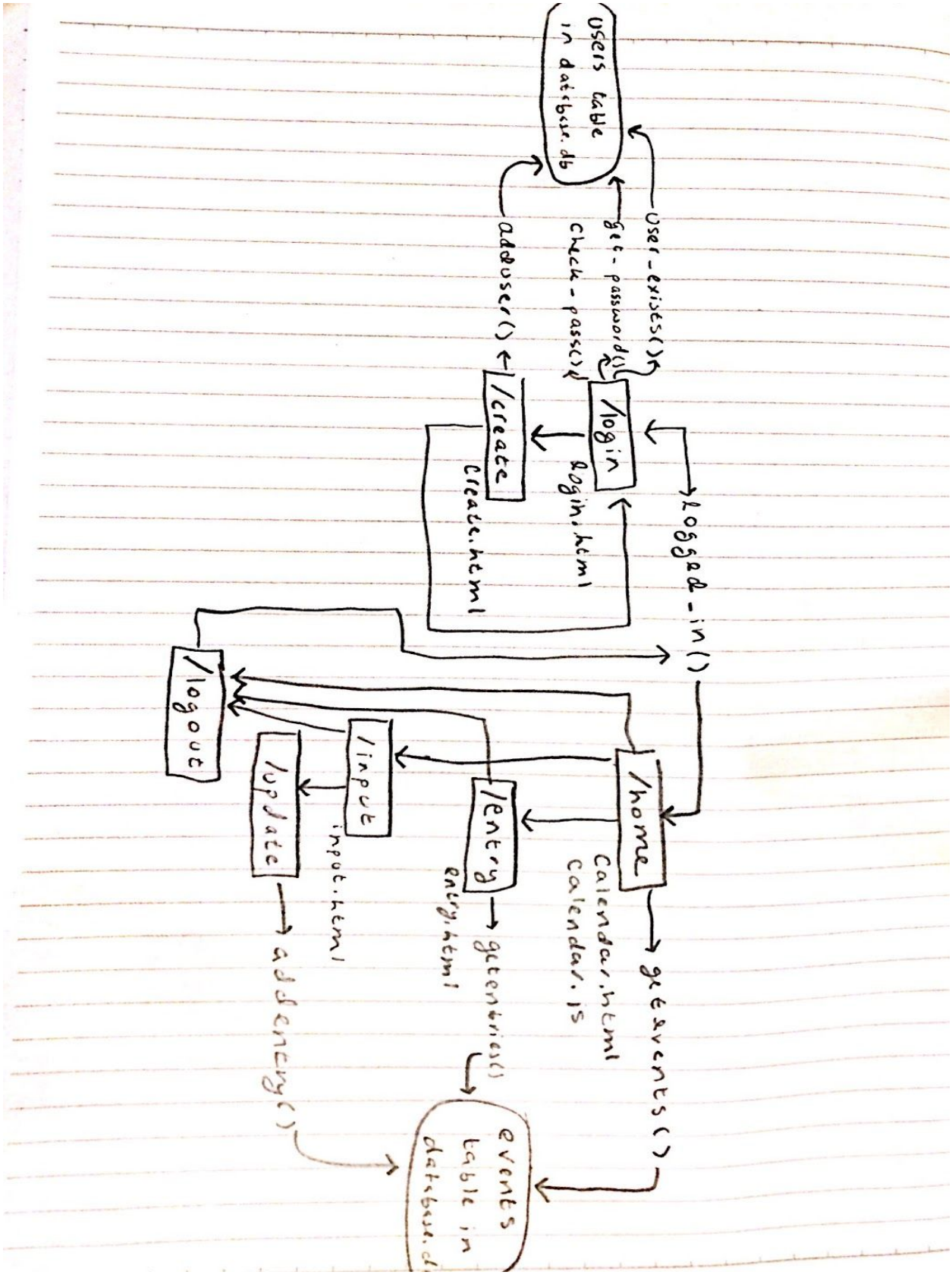
michelamar	051018	0	pic.jpg
bermet	051318	1	“This is my post”

Entry types: photo = 0, text post = 1, music = 2, event = 3

Site Map



Component Map



APIs
YouTube iFrame API to embed videos and Data API to search for videos

Roles

Michela	Back End: app.py (flask app), api, deployment & all things digital ocean
Tina	Back End: app.py (database) Front End: Input.html; entry.html
Bermet	Front End: frame.html, calendar.html, calendar.js, login.html, create.html

Timeline

- Log in system backend— 5/18
- Database stuff — 5/18
- Login system front and back end — 5/26
- Calendar home page — 5/26
- Creating entries, diff entry types — 6/1
- Adding events to calendar — 6/1
- Creating entry page — 6/8
- Adding video embedding to entries — 6/8
- Designs — 6/12