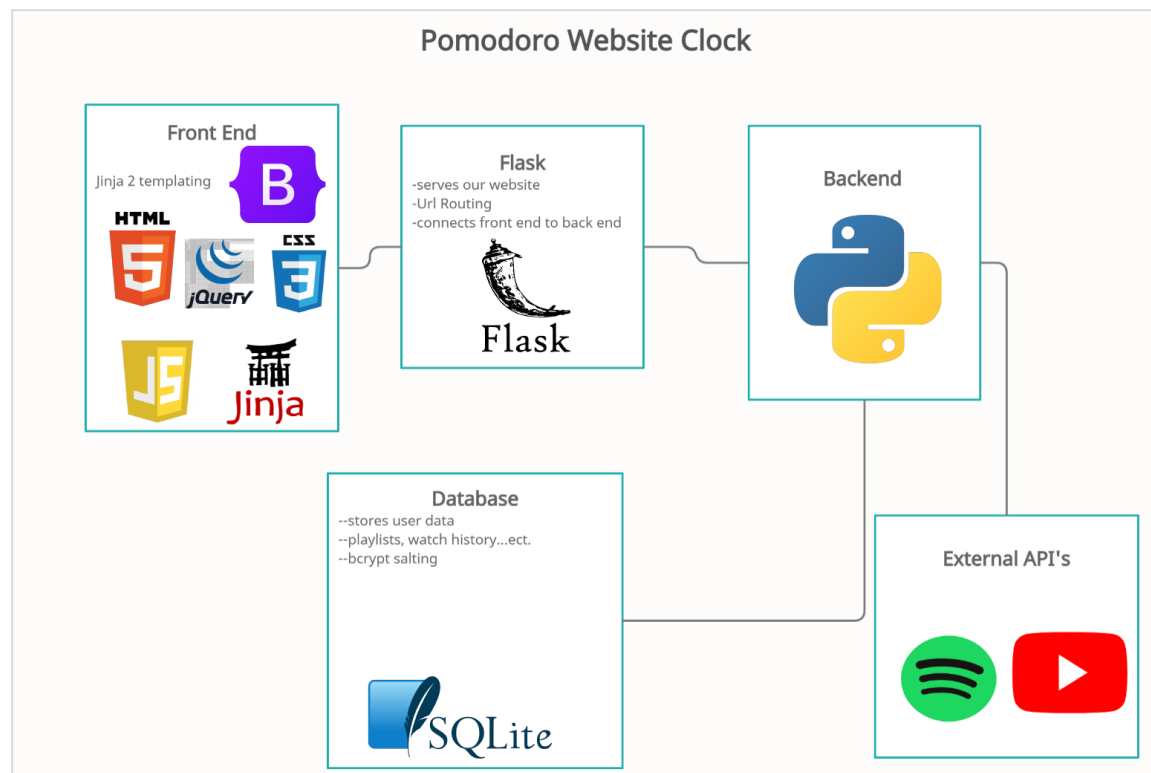


## **Revised Project Features List**

- (3) Registration and Login Pages using bcrypt
  - Registration and login pages for users to create their accounts and to access their saved playlists, videos, timer settings, etc.
- (1) Customizable Timers
  - Able to change the timers lengths to allow for different work/rest cycles
- (2) Linking Spotify and Youtube API's integration
  - Embedding multiple streaming services to include at least Youtube and Spotify in order to allow the user to choose their music. Tier 2 goal of utilizing the platform's API's for further configuration.
- (4) Data collection and User data saved
  - Allow returning user to return to the site with previous settings to include custom timers and music
- (7) Personalization Options (background etc)
  - Simple customization to allow the user to personalize the site a little more for viewing pleasure
- (6) Administrative Rights
  - Allow privileged users to access additional settings such as managing the user database. (perhaps to delete inactive users)
- (5) Animation and Presentation from "work to chill" shift and back.
  - When a user transitions from work to rest mode, the screen will shift back to the application to indicate that the work or rest period has ended and the next timed section is beginning.

Architecture Diagram

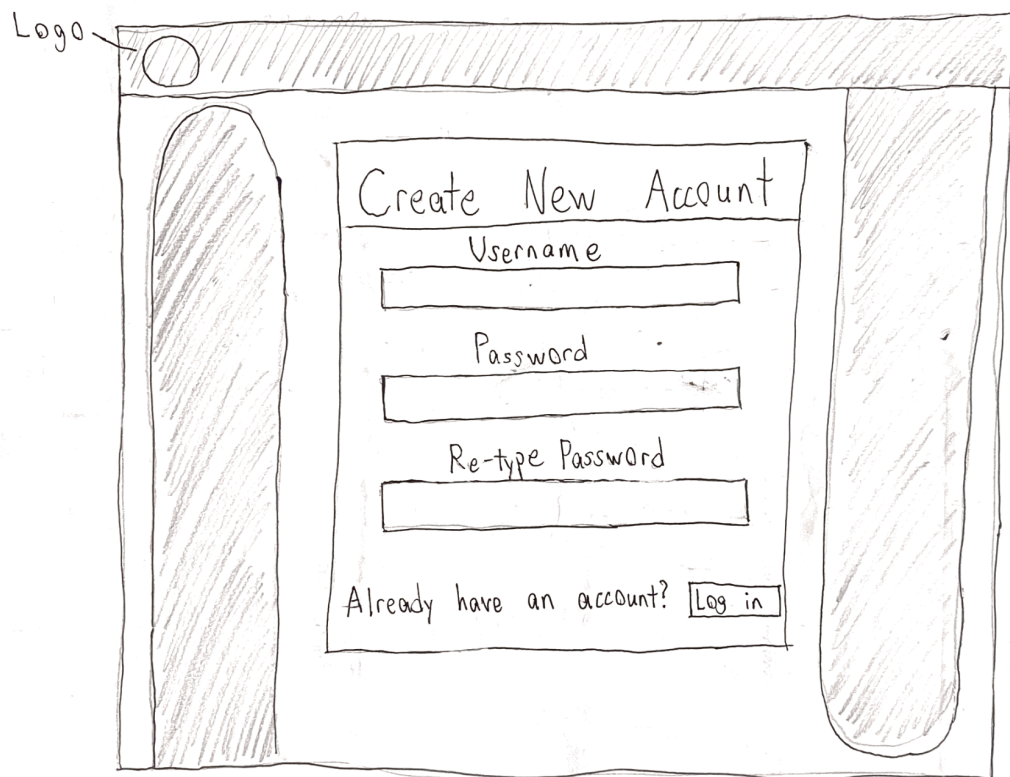


#### Front End design

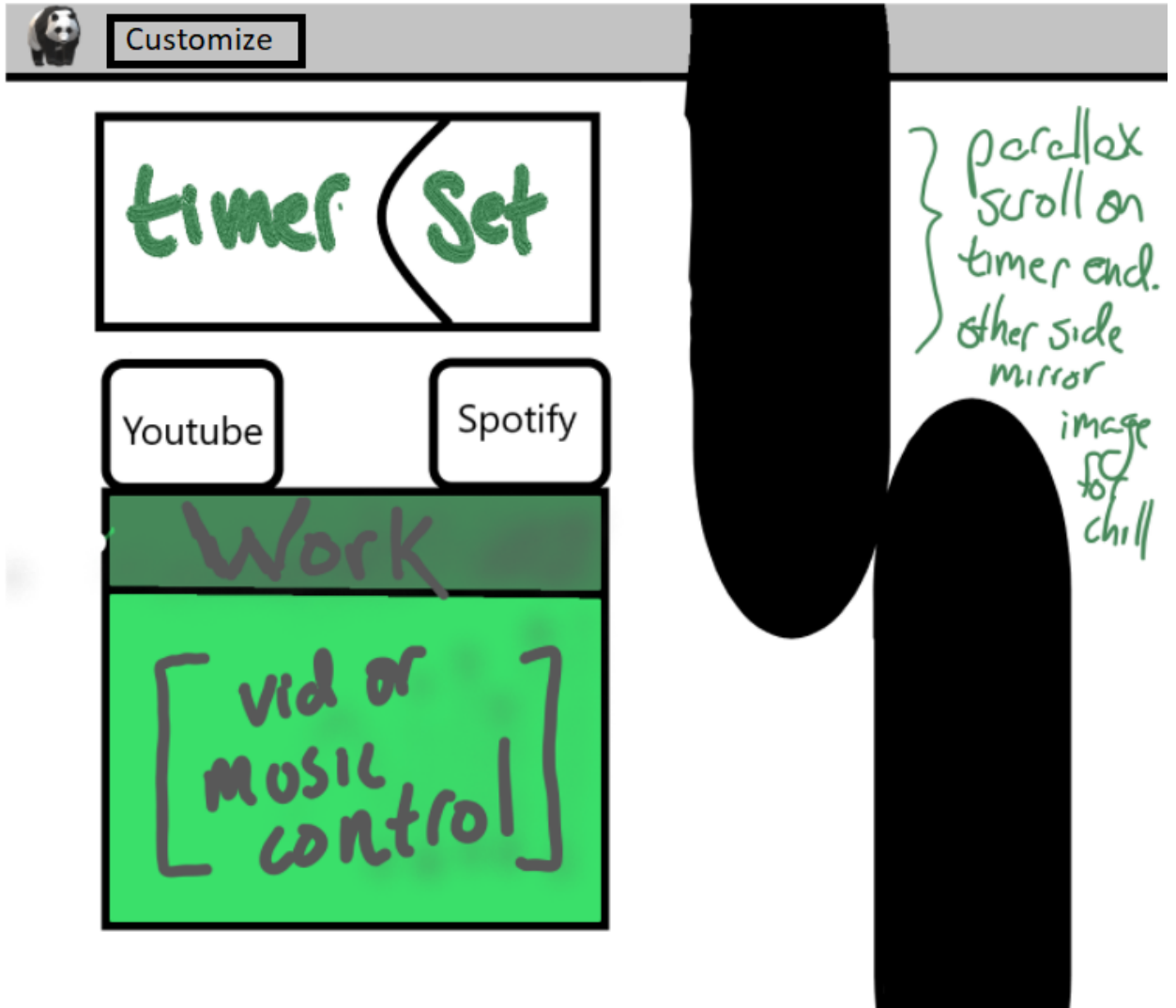
- Login Page:



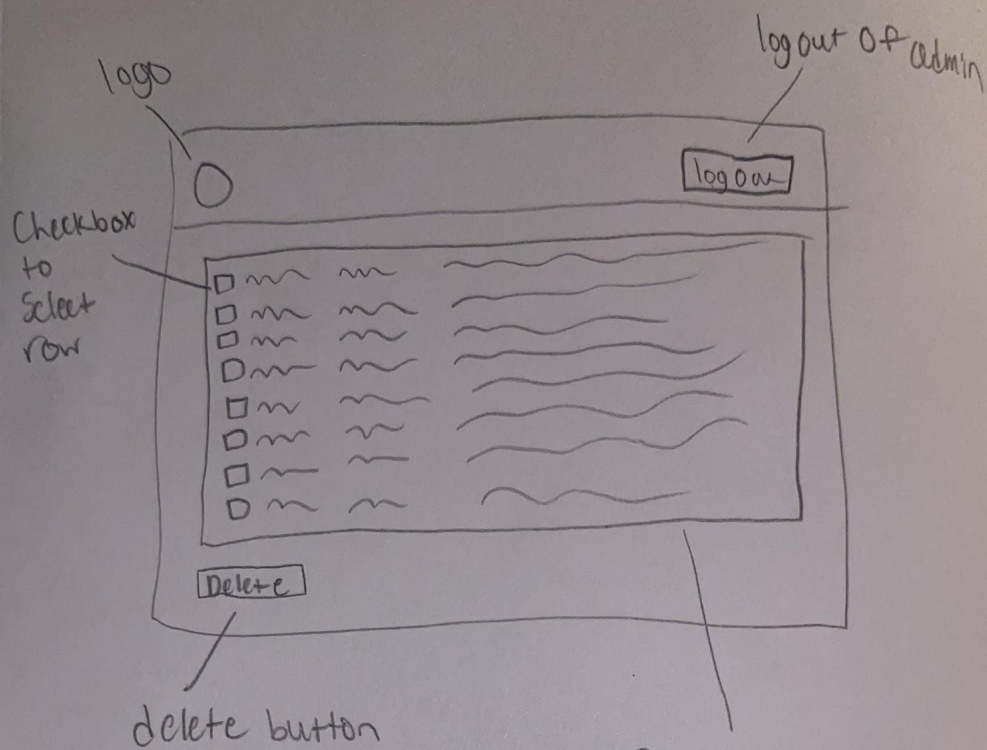
- Registration Page:



- Main Landing Page:



- Admin Page:



Scrollable table  
with username, name  
last logged in recently  
and other info

## Web Service Design

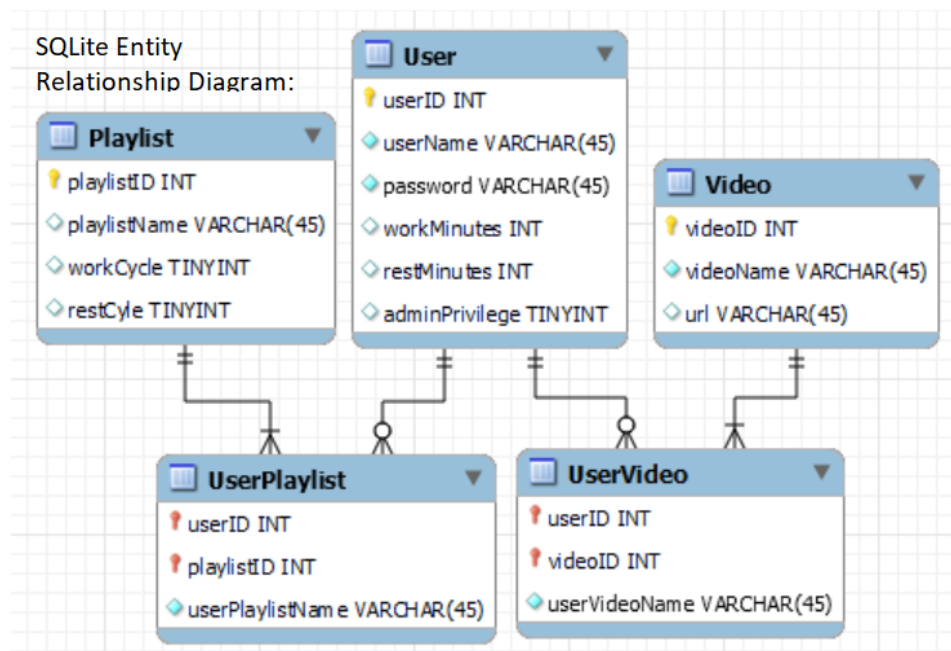
If your application is using Web Services via APIs, this deliverable should list the Web Service being used along with a description of the Web Service's API including the data being passed to and received from the API.

**Spotify:** Using SpotiPy, a python library for the spotify API, our application receives a user token for authentication and is directed to the spotify login where the API receives the user login. Then, our application receives a list of the user's playlists and playlist IDs for each. In order to display the embed of a playlist, Spotify receives the user's choice of playlist through the playlist ID/embed link.

**Youtube:** Utilizing Youtube's IFrame API, the application is designed to take in "embedded" URL's or "watch" URL's as well as utilizing the full functionality of the API to include custom control, auto playing and pausing. The only data being saved from the API is the video ID in order to be saved for the user.

## Database design

SQLite Entity Relationship Diagram:



## Challenges

- Identify at least 3 challenges and/or risks to your project at this point.
  - Idealistically utilizing our full database capabilities(getting login, spotify, etc. data into database)

2. Getting both spotify and youtube APIs fully functional, and playing well with each other.
  3. Integration of all parts (merge conflict resolution)
- What is your backup/risk mitigation plan in case you are unable to resolve the challenge(s)?
    - Change the bounds of the requirements, to be more manageable.
    - Merge conflict happening as a group meeting to discuss all changes
    - If the spotify API does not function as required with user playlists, then the user can input an embed link to a playlist instead.
    - Database requirements could be scaled down to bare necessities if required

### Individual Contribution

- Aiden -
  - Set up a working customizable timer with a start and pause/resume button. The resume/pause button changes based on what is currently needed. Separated the work and chill sides into their own javascript and html. And the chill side is now completely reactive, meaning whatever time you put into the work clock under an hour, it will auto fill the chill side with 60 - workTime. Autostart of the other sides clock and be enabled, but not sure if we want to implement that in the final version.
  - <https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-015-06/commit/0665562c29d9c499618eb485812ca12324b52f58>
- Daniel - Set up SQLite database and made Entity Relationship Diagram. Worked on routing logic for registration and login pages. Also set up bcrypt for encrypting passwords and checking password input on the login. Set up admin page to display database contents and delete inactive users.
  - <https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-015-06/commit/b3f0a7e68c909a879d615c97c597d7955d952cb6>
- Josh -
  - Scaffolded file structure and spun up project. Created animations for the website to include parallaxing background and foreground as well as timer entrance. Integrated Youtube API and working on saving videos for users.. Wireframed main page, and CSS'd neumorphic internal scroll for video selection and glass overlay for media player.
  - <https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-015-06/commit/7507614c67254d9e664ba3cea2287133a7271a52>
- Tea -
  - Connected Spotify API through Spotipy so that a user can log in and access the playlists saved in their library. The user can play these through a spotify embed and switch playlists. Currently working on making the playlist options a dropdown menu, duplicating the embed on both sides of the website, and caching the login data.
  - <https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-015-06/commit/dc2d11525f48d8f2d0c2302cc18c408e5b0c13a3>

- Uri - Architecture Diagram, artwork, CSS
  - Made the designs consistent. Integrated the navigation bar to create one unified navigation bar across the whole website (before this each page had its own navigation bar). Made the backgrounds the same in all pages. Made all of the sidebars the same (the main page and the login page had different sidebars). Hid password by default and added show password button.
  - <https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-015-06/commit/84c78e8b60d6045a7f438bbb391ad90390c6bef9>
  - <https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-015-06/commit/e17bcca3dac97b8e32936db588c378c07fc1401f>
  - <https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-015-06/commit/b2227e46688ec72c874b03400c906a16302e5534>
  -
- Yan - User customization. Users can choose other backgrounds of the website.
  - <https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-015-06/commit/c4e40dc424050c0e64f1708685d92b6f258924de>
- Project Management Board

