# System Design Document

Learning Task

Overview: The Learning Task is the first thing a regular user will see each day. It will introduce a pool of face-name pairs of the appropriate difficulty for the user's current level of progress. It will introduce each pair one at a time, and these face-name pairs will be used for the rest of the activities the user does on the same day.
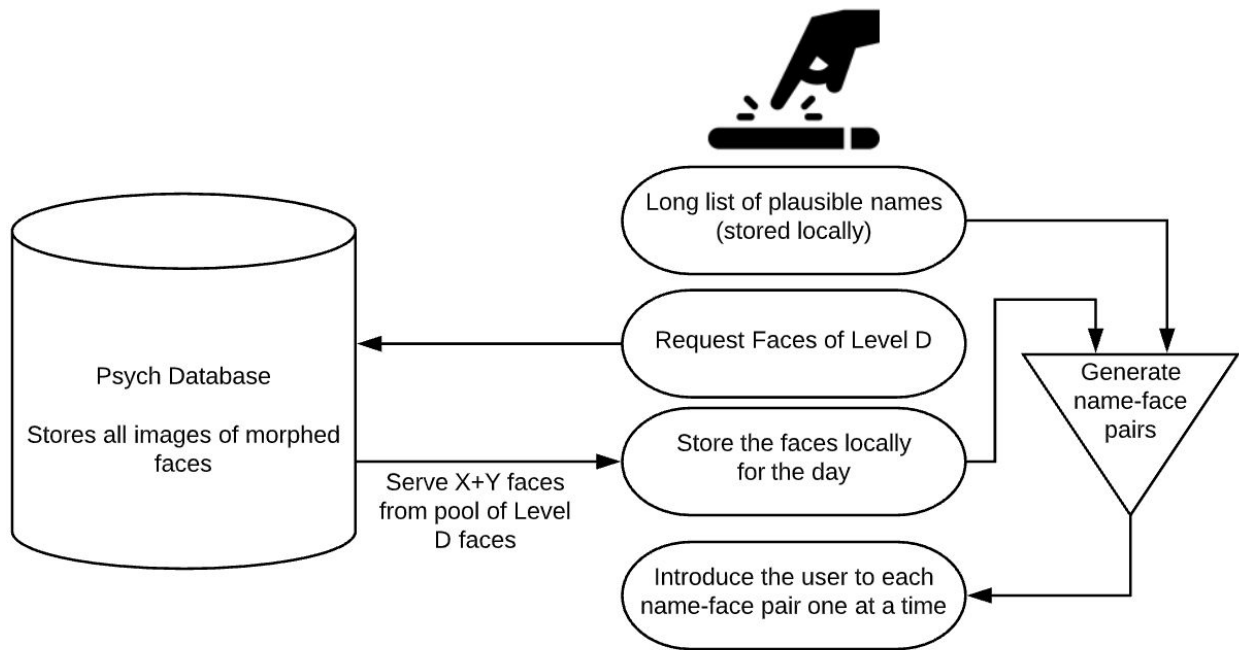

Note: The database should serve X+Y faces, in case the user wants to do "Who's New?" Where X is the base number of faces shown each day in the learning activity and Y is how many times the user should be able to repeat an activity. (Because we need one new face each time "Who's New?" is played.)

Disclaimer: The depicted method of pulling faces from the database each day is not set in stone. It might be more feasible to save a compressed version of the whole set of faces locally, largely dependent on how many faces will be used through the whole app.
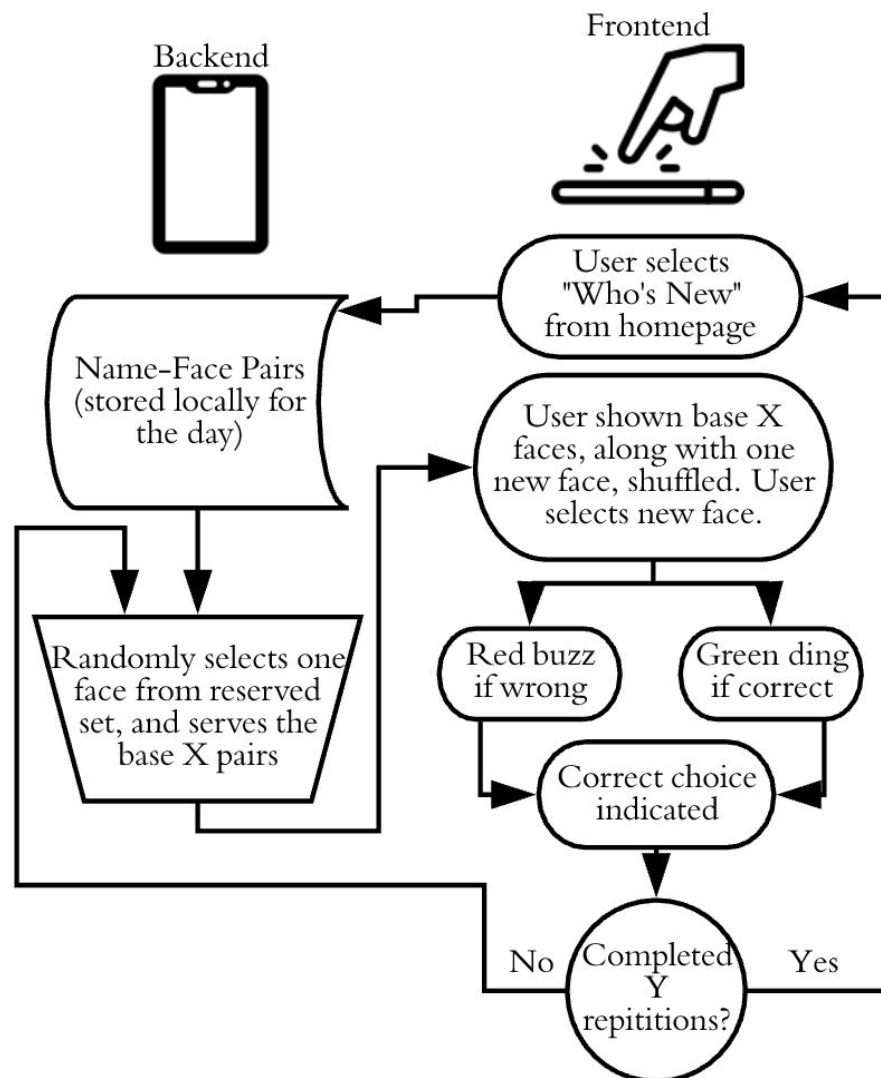
Practice Tasks

Name and Face: This task serves to help the user become more fluent at individuating faces of another race. The user will be shown a name from the set of name-face pairs they were given for the day. The user will then be shown X faces, and be asked to pick the face matching the given name. X is a function of the user's progress level.
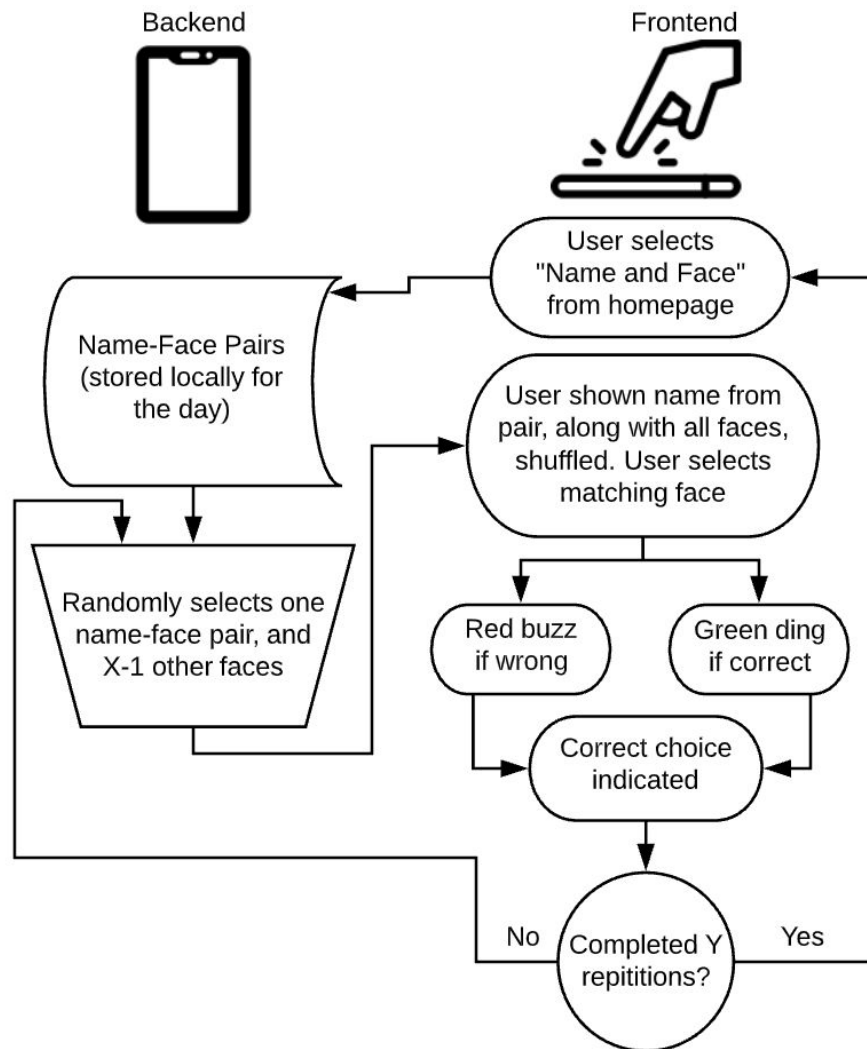

Note: The "X-1 other faces" that are selected for this activity should be the rest of the base set shown in the learning activity, that were not selected to have their name used.

Who's New: This task serves to help the user become more fluent at individuating faces of another race. The user will be shown X+1 faces, where X is the faces they were presented in the initial Learning Task for the day, and the extra face comes from the set reserved for this task but not yet shown to the user. The user is then asked to identify the new face, and given feedback on whether or not they were successful.

Backend

Frontend

Name-Face Pairs (stored locally for the day)

User selects "Who's New" from homepage

User shown base X faces, along with one new face, shuffled. User selects new face.

Randomly selects one face from reserved set, and serves the base X pairs

Red buzz if wrong

Green ding if correct

Correct choice indicated

Completed Y repititions?

No   Yes

Note: Reserved set was pulled from psych database during the Learning Task and set aside for use in this task. Upon repetition, it should be assured that a new face is selected. Probably just remove each face from the reserved set once it's been used.

Memory-Match: This task serves to help the user become more fluent at individuating faces of another race. The user will be shown an NxM grid of cards. (Where N and M are functions of the user's progress level. N*M should not exceed 2X). The fronts of the cards (which are facedown) will be the base X faces shown during the Learning Task. The backs of the cards should be pictures of me in an orange tuxedo. The user will then select two cards a time. If the faces match, they will remain upright. If not, then the cards will revert to face down. This will continue until the entire board is solved.
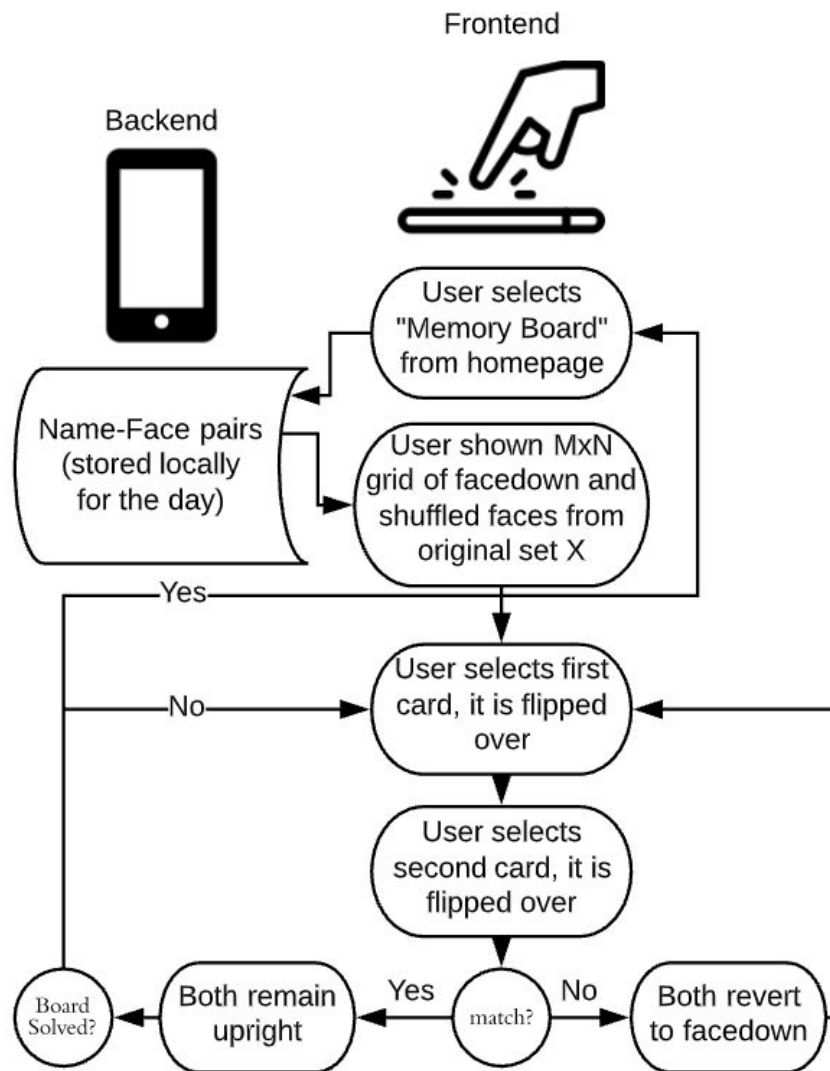
Note: I think this one is pretty subject to change. For example: should they get a sneak peak of the cards before being flipped over? Should there be a time limit? Etc.

Shuffle: This task serves to help the user become more fluent at individuating faces of another race. The user will be shown the set (possibly a subset?) of the original X faces shown in the Learning Task this day. The user will be given T amount of time to study it, then the user will be shown a shuffled list of the same faces. The user will be asked to sort the list into the correct order. Correct and incorrect slots will be indicated.
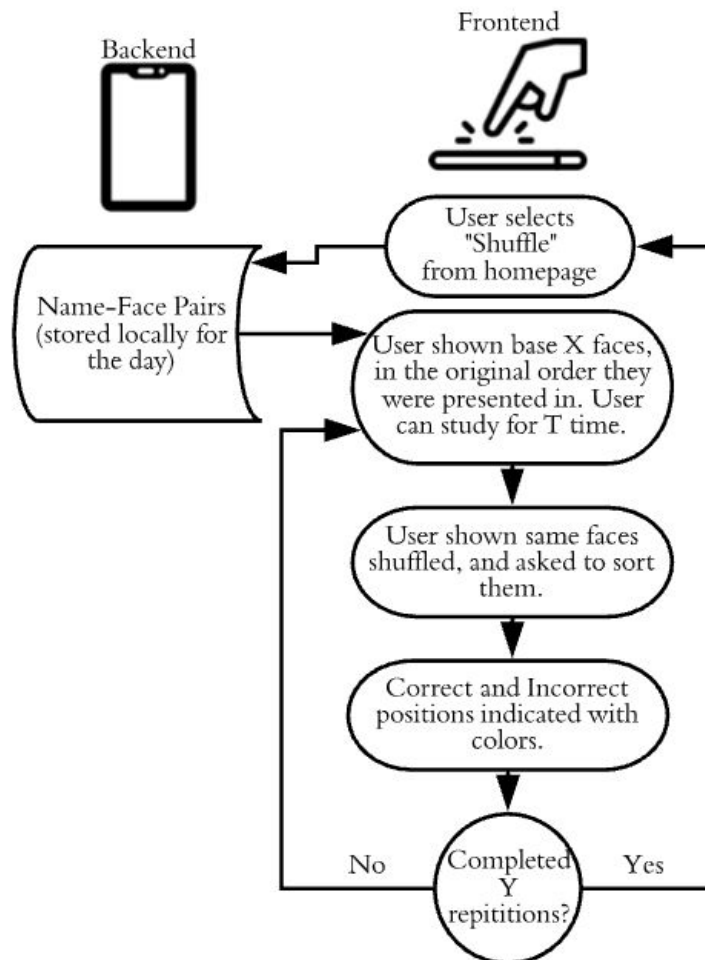
Assessment Tasks

Forced Choice: Users are shown a face which they must memorize. They are then shown a set of faces, from which they have to choose the face first face they were shown.

```
                              User selects
                              Forced choice or
                              is taken there at
                              end of module
                                   │
                                   ▼
   Face-name pairs   ──────▶   User is shown one
   stored locally                   face
                                   │
                                   ▼
                              User shown
                              multiple faces, one
                              of which was
                              shown before
                                   │
                                   ▼
                              Choose which
                              face was shown
                              before
                                   │
                                   ▼
                                Correct?
                          Yes   │     │   No
                          ┌──────┘     └──────┐
                          ▼                    ▼
                   Positive feedback    Negative feedback
                          │                    │
                          └────────┬───────────┘
                                   ▼
                              Display module
                              completion status
```
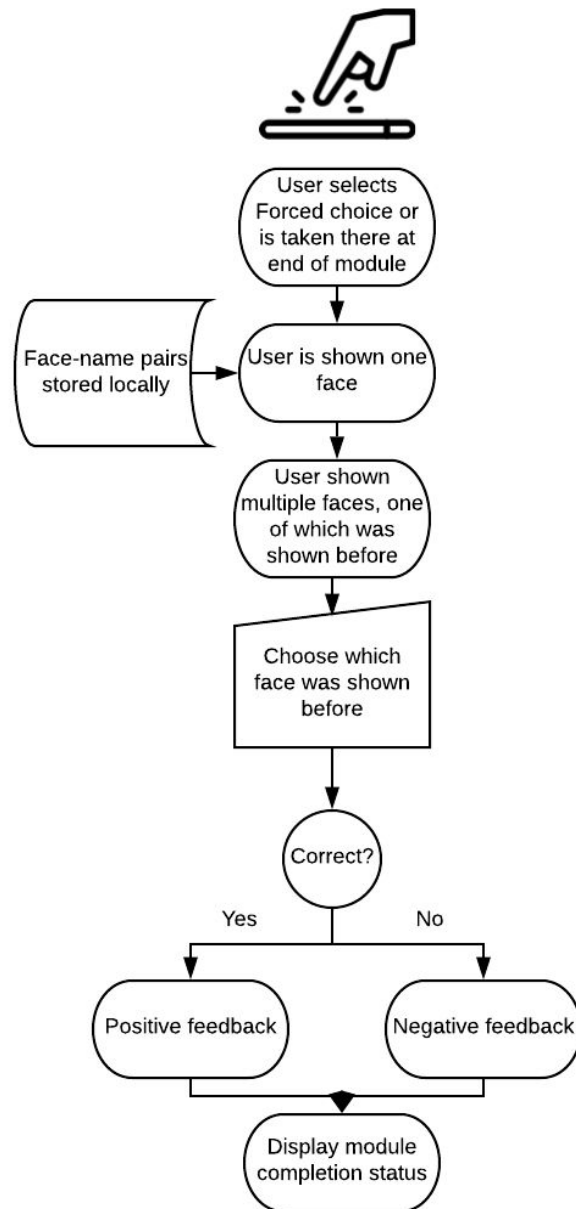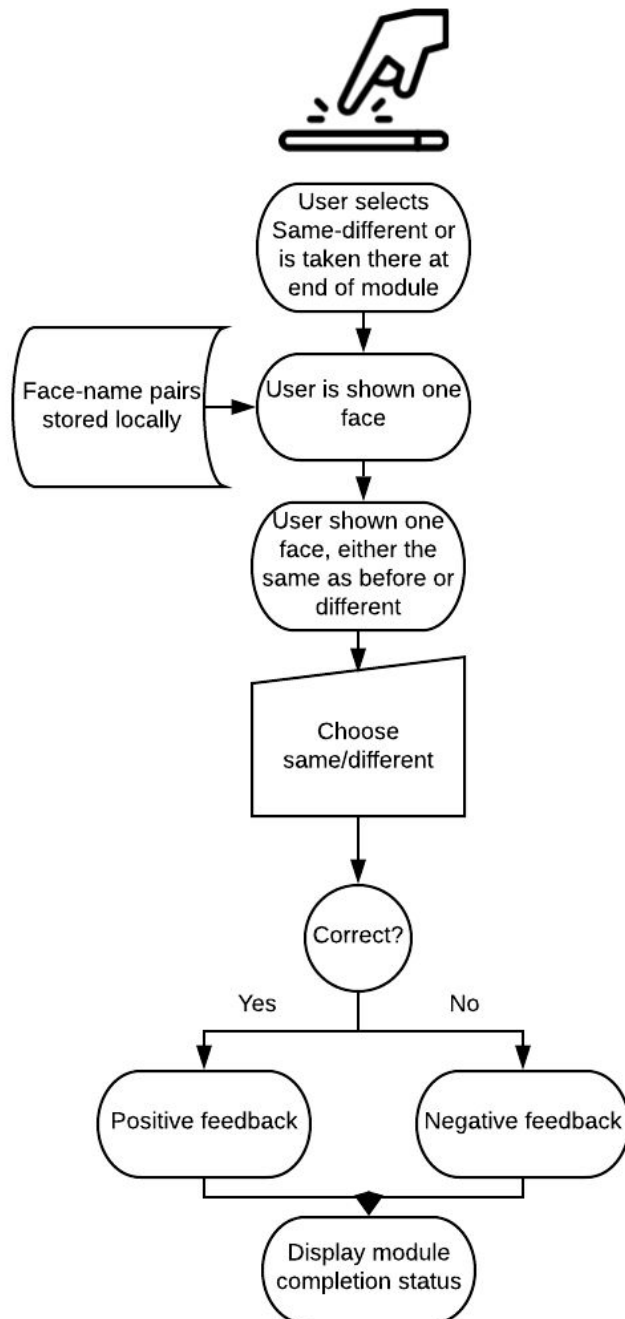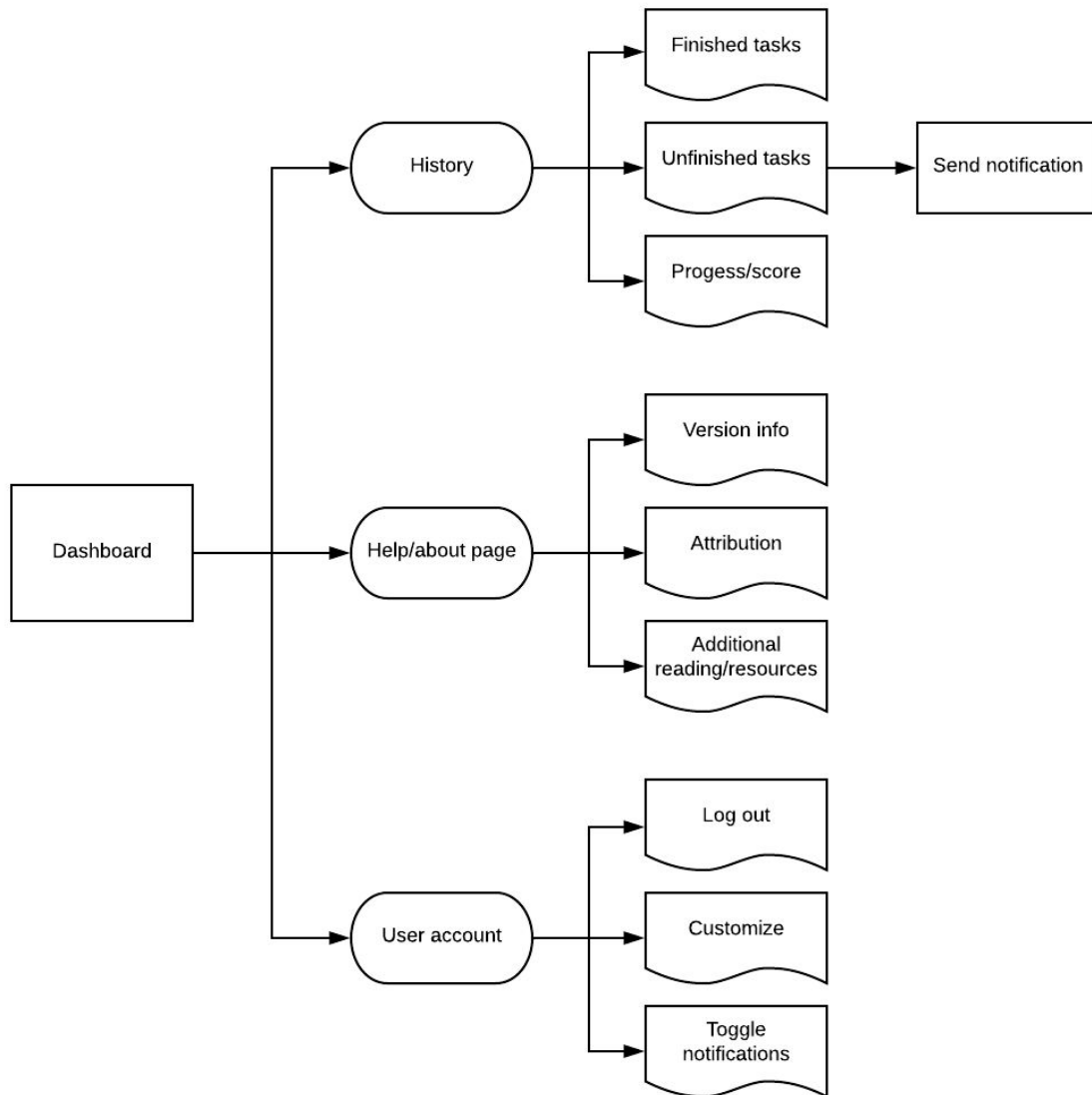
<u>Same-Different</u>: Users are shown a face which they must memorize. Then they are shown a
blank screen for a number of seconds. They are then shown another face, which they must decide
is the same as the first.

User selects
Same-different or
is taken there at
end of module

Face-name pairs
stored locally → User is shown one
face

User shown one
face, either the
same as before or
different

Choose
same/different

Correct?

Yes   No

Positive feedback   Negative feedback

Display module
completion status

DashBoard(History & Notification)

Backend and server connection

An API connected to the server storing results will be made available. The app will connect to the API at three primary points, representing three different user facing tasks. The first is during user registration, where the app will send the user data to the API, which will create an entry for the user in the database. After task completion (training or assessment) a summary will be created and sent to the API, where the data will be stored in the database. Finally, during user sign in the app will query the API to check for the most up to date record stored for that user. If there is a record more recent than what the app knows of, this means the user logged in elsewhere and data needs to be synced. Using the data from this request the app will complete the sync. This portion allows for users to maintain progress across devices.

The CU Psychology department will not use the API to pull data, but will instead directly query the database to get user information. Due to the scope of this project, support may be given to facilitate this process, but no additional application or API endpoint is planned.