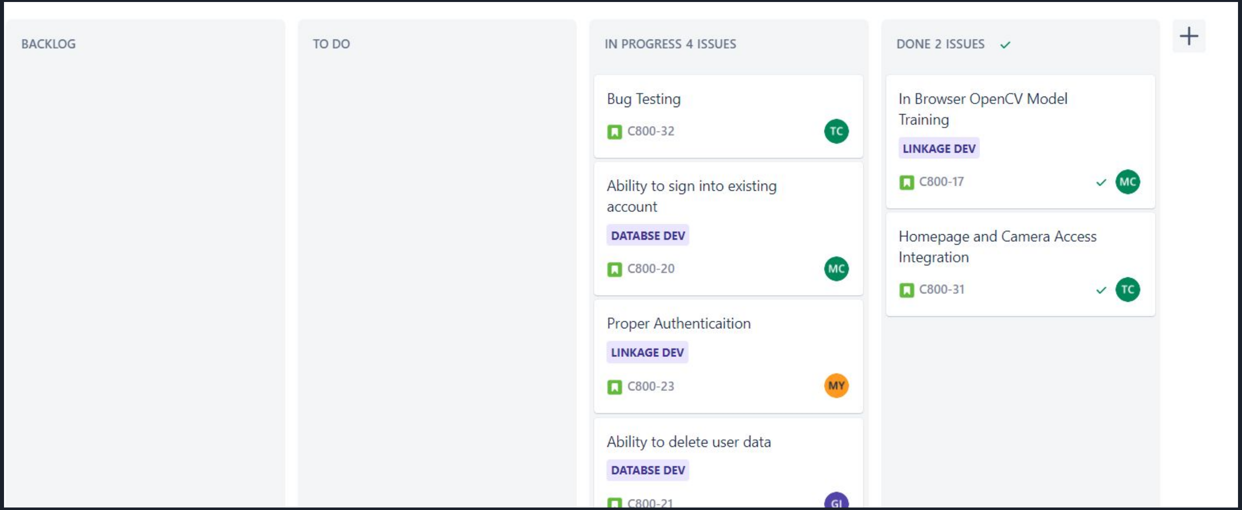
CFYF

CANT FORGET YOUR FACE

By: Gregory James, Matthew Ceriello, Michael Yon and Tyler Curnow

Tools Used

- Jira (project tracker)
  - Purpose: To track the progress of the development of our application and keep us on the same page regarding what we should be working on
  - Rating: 2/5
  - We didn't use Jira very often and discussing what we needed to do weekly was much easier. Jira is a good tool, but for such a small and focused project it didn't feel very useful.



BACKLOG

TO DO

IN PROGRESS 4 ISSUES

Bug Testing
C800-32 TC

Ability to sign into existing account
DATABSE DEV
C800-20 MC

Proper Authenticaition
LINKAGE DEV
C800-23 MY

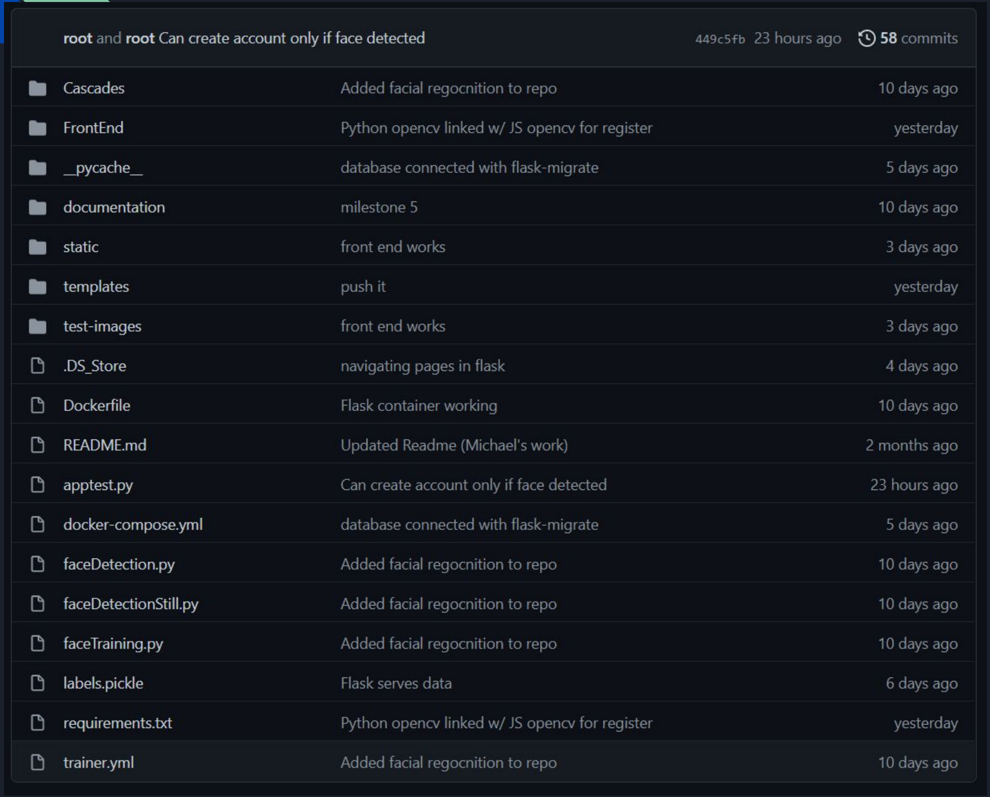
Ability to delete user data
DATABSE DEV
C800-21 GI

DONE 2 ISSUES ✓

In Browser OpenCV Model Training
LINKAGE DEV
C800-17 ✓ MC

Homepage and Camera Access Integration
C800-31 ✓ TC

Tools Used



root and root Can create account only if face detected 449c5fb 23 hours ago 58 commits

Cascades	Added facial regocnition to repo	10 days ago
FrontEnd	Python opencv linked w/ JS opencv for register	yesterday
pycache	database connected with flask-migrate	5 days ago
documentation	milestone 5	10 days ago
static	front end works	3 days ago
templates	push it	yesterday
test-images	front end works	3 days ago
.DS_Store	navigating pages in flask	4 days ago
Dockerfile	Flask container working	10 days ago
README.md	Updated Readme (Michael's work)	2 months ago
apptest.py	Can create account only if face detected	23 hours ago
docker-compose.yml	database connected with flask-migrate	5 days ago
faceDetection.py	Added facial regocnition to repo	10 days ago
faceDetectionStill.py	Added facial regocnition to repo	10 days ago
faceTraining.py	Added facial regocnition to repo	10 days ago
labels.pickle	Flask serves data	6 days ago
requirements.txt	Python opencv linked w/ JS opencv for register	yesterday
trainer.yml	Added facial regocnition to repo	10 days ago

- Github (VCS Repository)
  - Purpose: To manage our project files across multiple developer computers
  - Rating: 5/5
  - Github is fast, free, and fully featured git ecosystem. It was essential to getting our program working on multiple computers.

Tools Used

- PostgreSQL (Database)
    - Purpose: Act as the database backend, managing user data.
    - Rating: 5/5
    - Very easy to configure inside of a Docker container and integrated directly with flask using SQLAlchemy

Tools Used

- Localhost (Deployment Environment)
  - Purpose: Where we are deploying our application.
  - Rating: 5/5
  - Deploying to localhost is simple and makes testing easy. It is possible to deploy this application to the cloud, but it would require many more features be implemented in order for it to work.

- Docker / Docker-Compose (Container)
  - Purpose: Because our application relies on a suite of libraries to run correctly, building one image with all of them installed and running our application inside of the container was the obvious choice.
  - Rating: 5/5
  - Building a custom image is easy and integrates so seamlessly with a git repository it’s almost comical. Having a consistent environment for our application to run in was crucial, and Docker makes it easy

Tools Used

- Tiny cloud
  - Purpose: Easy deployment of a shell for text editing
  - Rating: 4/5
  - Easy to deploy but hard to master and most features are blocked behind a paywall.

- Socket.io
  - Purpose: to stream video data between the client and server
  - Rating: 4/5
  - Very useful and not too difficult to configure, but the results are not always pretty

Tools Used

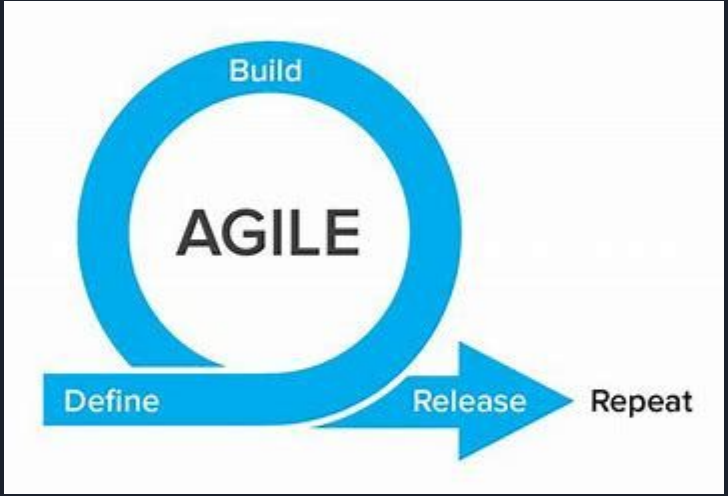
- VSCode(IDE)
    - Purpose: A multipurpose development environment
    - Rating: 5/5
    - VSCode's extension system is amazing. It allowed us to customize our IDE to suit our needs perfectly.

- Flask (Framework)
    - Purpose: The backend web framework we use to host/serve our web pages, integrate with our app layer, and query our database. It uses python which
    - Rating: 4/5
    - Very easy to pick up and learn, which was nice because we didn't learn it in this class. Integrated seamlessly with our application because its all in python. Can get a bit messy at times.

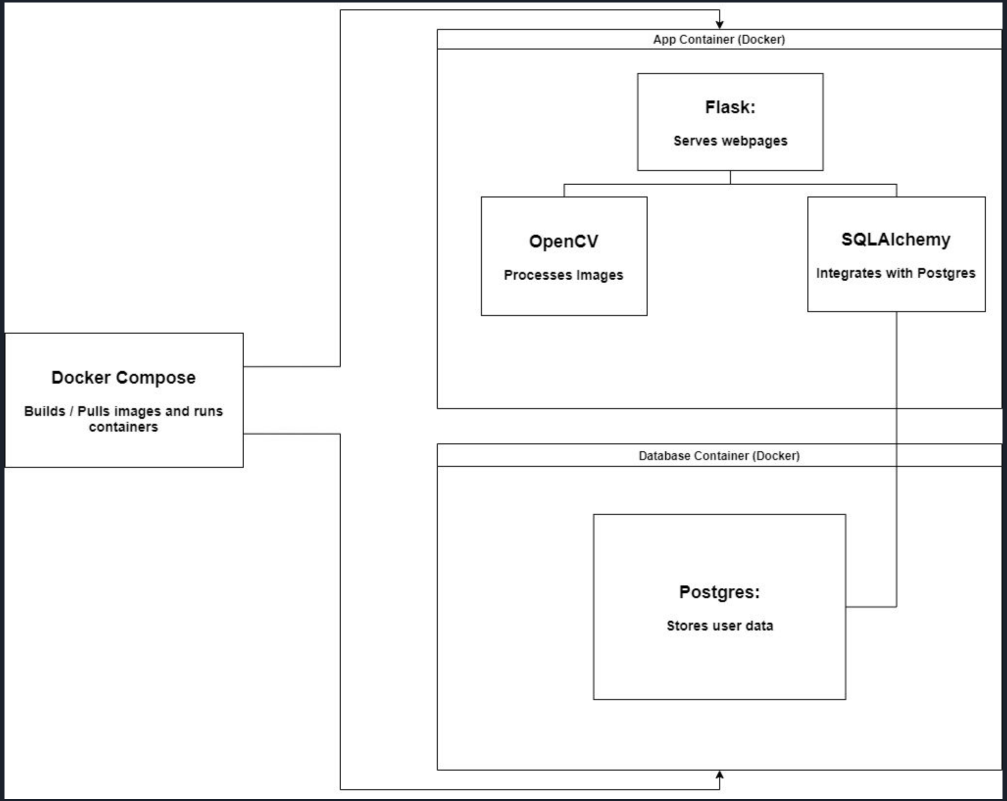
- OpenCV (Main library)
    - Purpose: Allows us to process images
    - Rating: 4/5
    - Amazing technology in a very nice package. Requires a lot of other libraries to function correctly which bloated our container.

Methodologies

- Agile
  - Made the most sense as we needed to adapt our program to what we would eventually learn we were capable of
  - Prioritizing smaller pieces of working software allowed us to figure out what would work and what wouldn’t



Architecture Diagram



Challenges

- Learning Flask
- Coming up with an architecture that works
- Learning OpenCV
- Integrating all of the small pieces we developed independently

Demo Time

If live demo doesn't work:

https://youtu.be/KBbrVDPI-4Y