Show Me a Funny Face

Dylan Finn, Jennifer Gulmohamad, Loyal Greene, Annie Zeng

Outline

- Motivation
- Proposed Idea
- Similar Projects
- Tools and Technologies
- Implementation
- Demo
- Summary/Overview
- Future Directions/Considerations
- Conclusion

Motivation

- Autism Spectrum Disorder (ASD) is a developmental disability caused by differences in the brain that affect how these people learn, communicate, and interact with others
- According to the CDC, the ASD rates in the US increased from 1 in 150 (2000), 1 in 54 (2016), and to 1 in 44 (2018)
- Common symptoms
  - Restrictive/repetitive behaviors
  - Challenges with social interactions/situations/communication
    - Little to no eye contact
    - Facial expression recognition/display difficult
      - Sometimes incorrect expression made in social situation

Proposed Idea

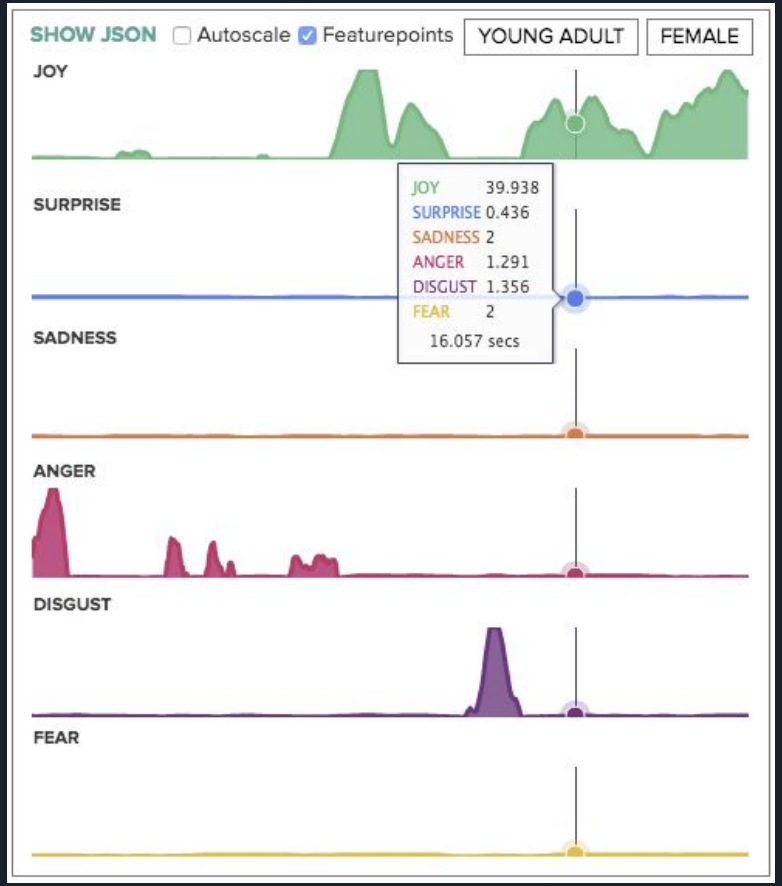
- Create application to help people with autism (specifically children)
  - Identify facial expressions
  - Locate and understand social situations
  - Practice facial expressions
- Webcam module that will identify facial expressions in real time
  - Train facial recognition model for webcam module

Similar Projects

- Kairos
- FaceReader
- MorphCast
- Paul Ekman Group

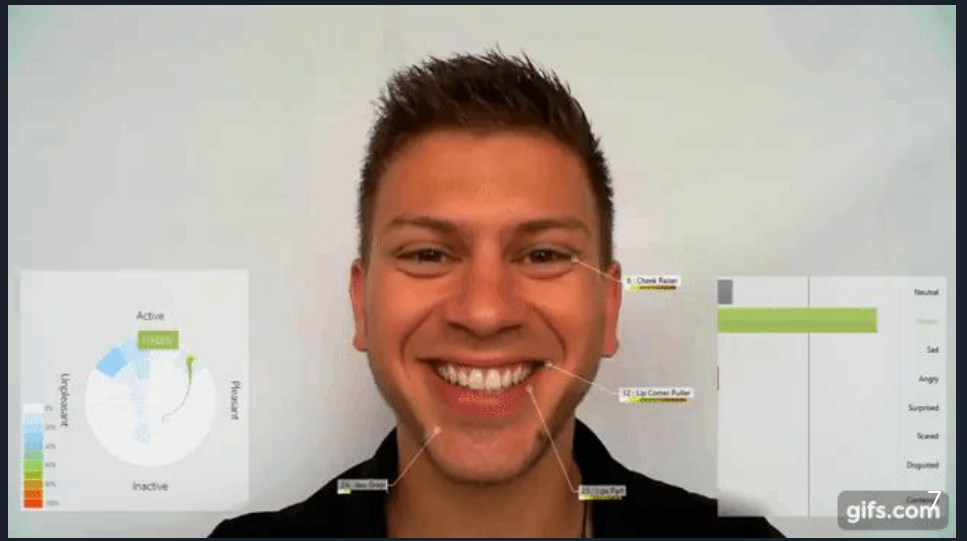
Kairos

- Face Recognition using cloud API
- Pricing ranges from $19-$499/month
  - With additional features depending on plan
- Several demos/API examples made available on Github
- Creates JSON responses using a script
  - Emotion charts
  - Facial feature points
- Webcam module possible



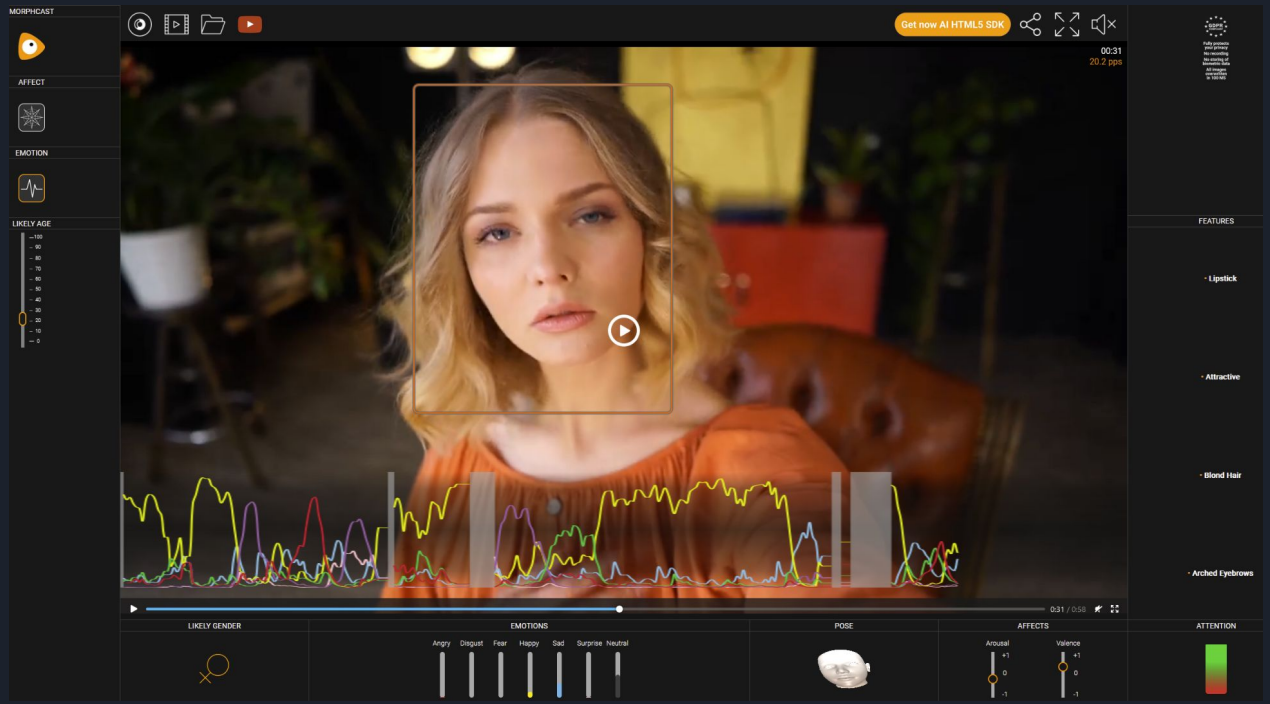
FaceReader

- Facial expression analysis
  - Classified into happy, sad, angry, surprised, scared, disgust, neutral
  - Custom possible
- East-Asian and baby models available
- Additional modules possible
- Mostly for research
  - $2,420—$10,340/year
  - Most cited facial expression recognition software



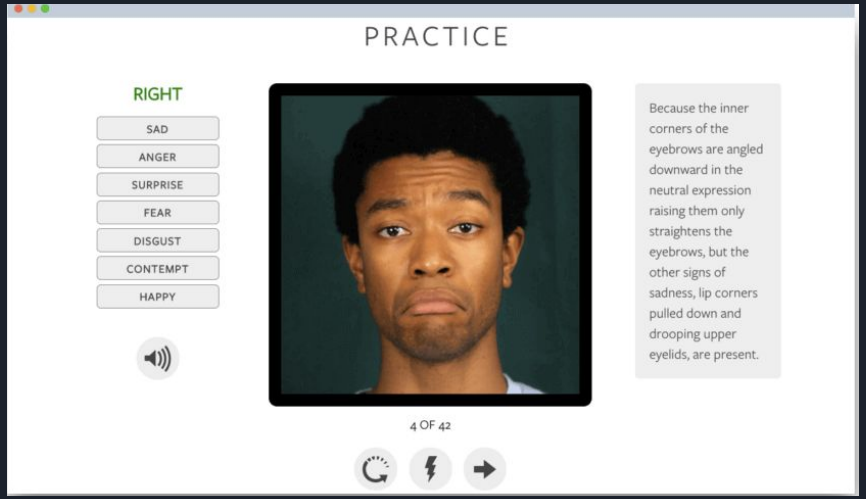
MorphCast

- Interactive videos based on viewer reaction/expressions
- AI HTML5 SDK also available (demo possible with own webcam)
  - JavaScript
- Pay as you go (dependent on amount and average duration of views)
  - Costs may vary from month to month



Paul Ekman Group

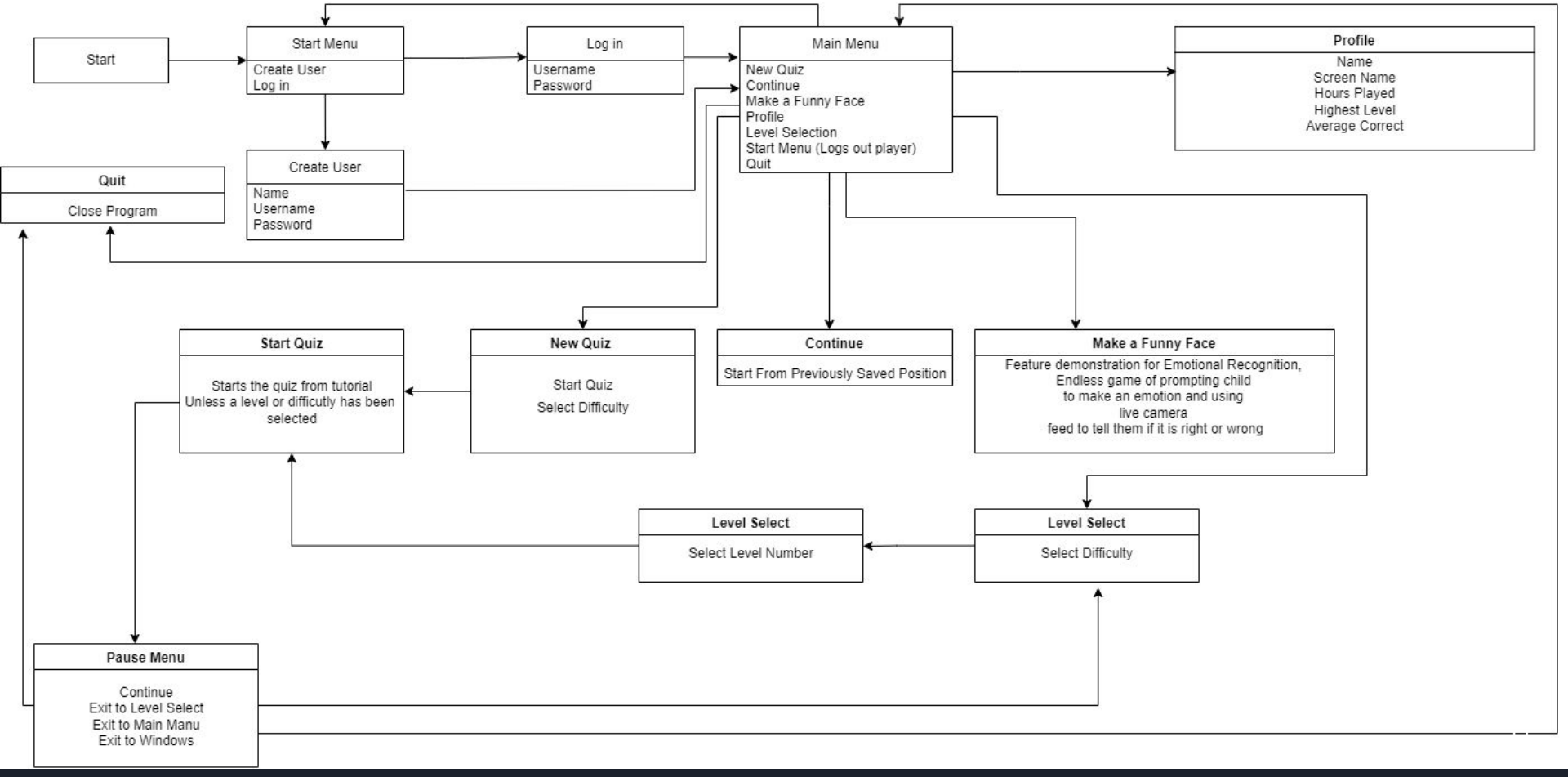
- Three different subscription packages for training
  - $119-$299
- Created by Dr. Paul Ekman
  - Well known for his research
  - Research used to identify seven universal facial expressions
    - Classifications in artificial intelligence, deep learning, neural networks utilize this research
- Closest to our project usagewise



Show Me a Funny Face!

- Specific to autistic needs
- Focus not only on facial expression detection
- **Develop an app to help learn facial expressions**
  - Easily accessible
  - No paywall
- Practice facial expressions with live feedback

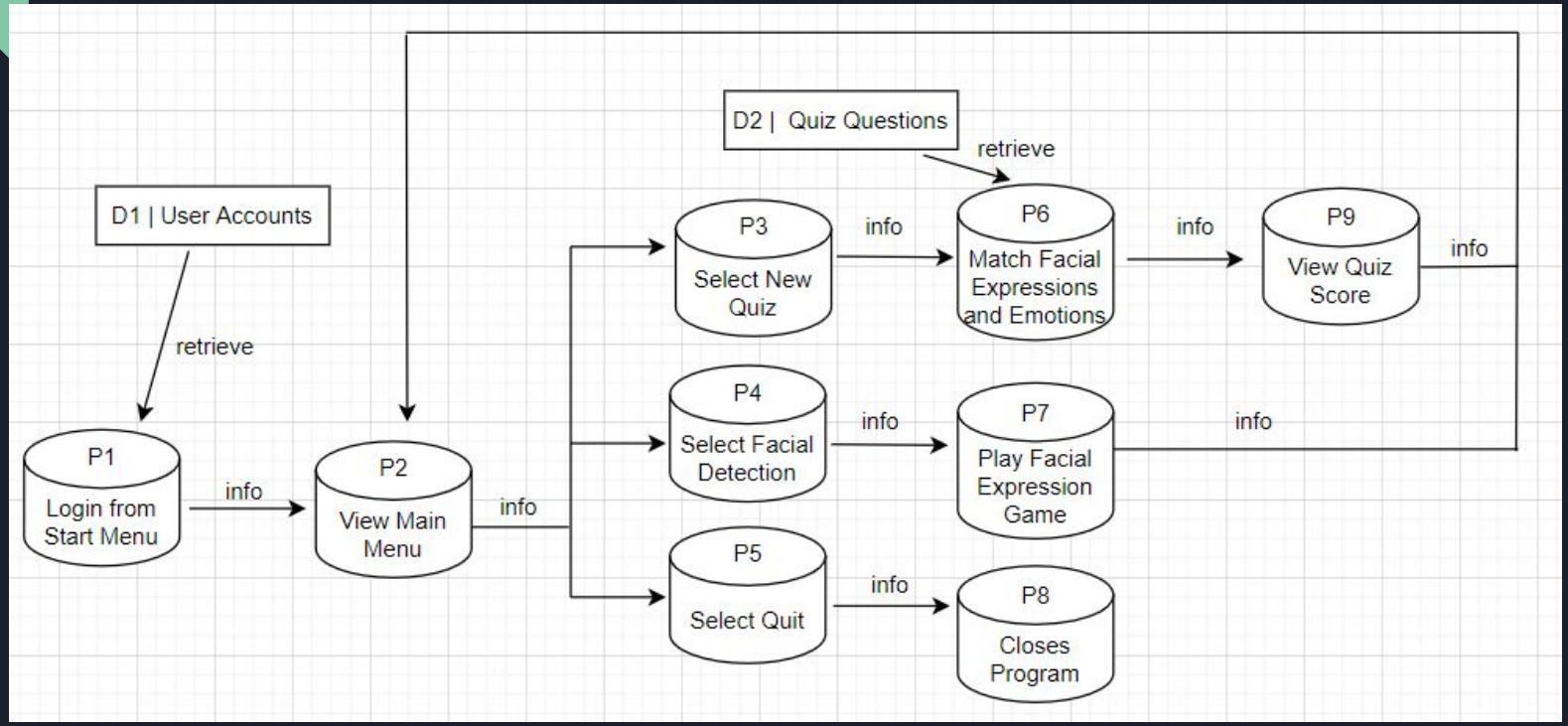
Menu UI Diagram



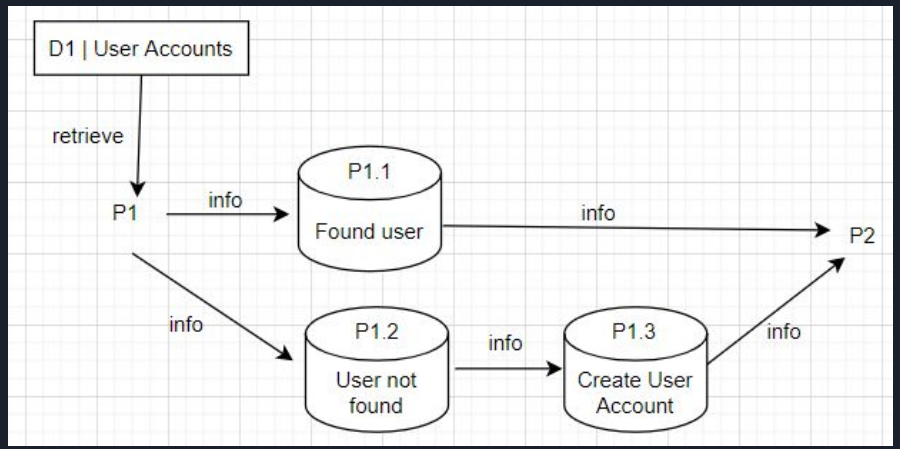
Tools and Technologies

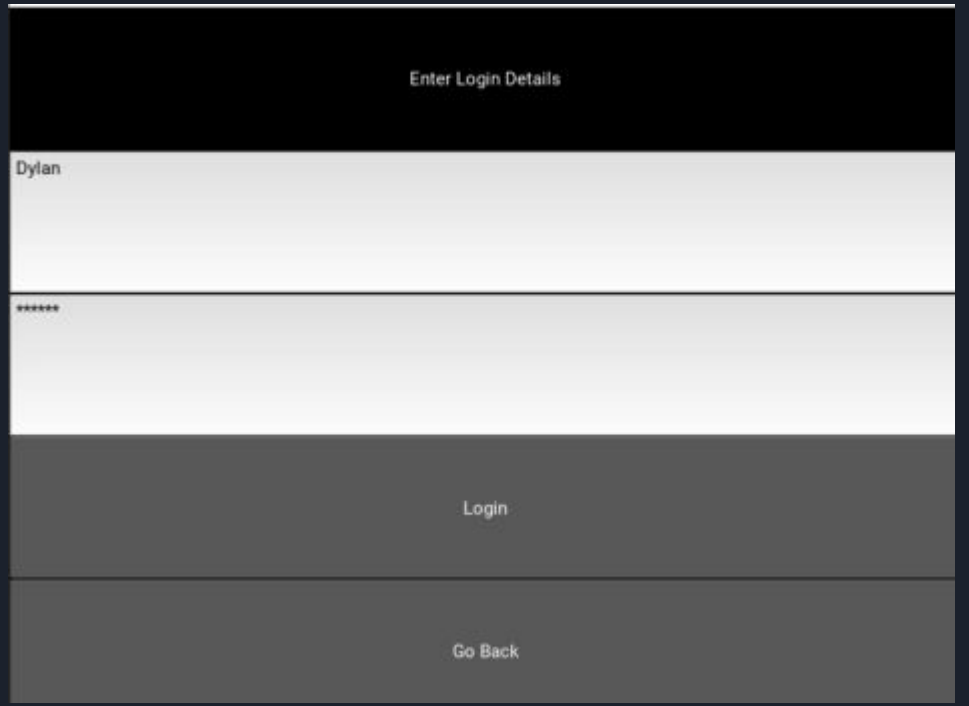
- Tensorflow and Keras
  - Machine learning
- Kivy
  - FrontEnd Framework
- OpenCV
  - Computer vision
- Database system
  - Local collection of JSON files

DFD

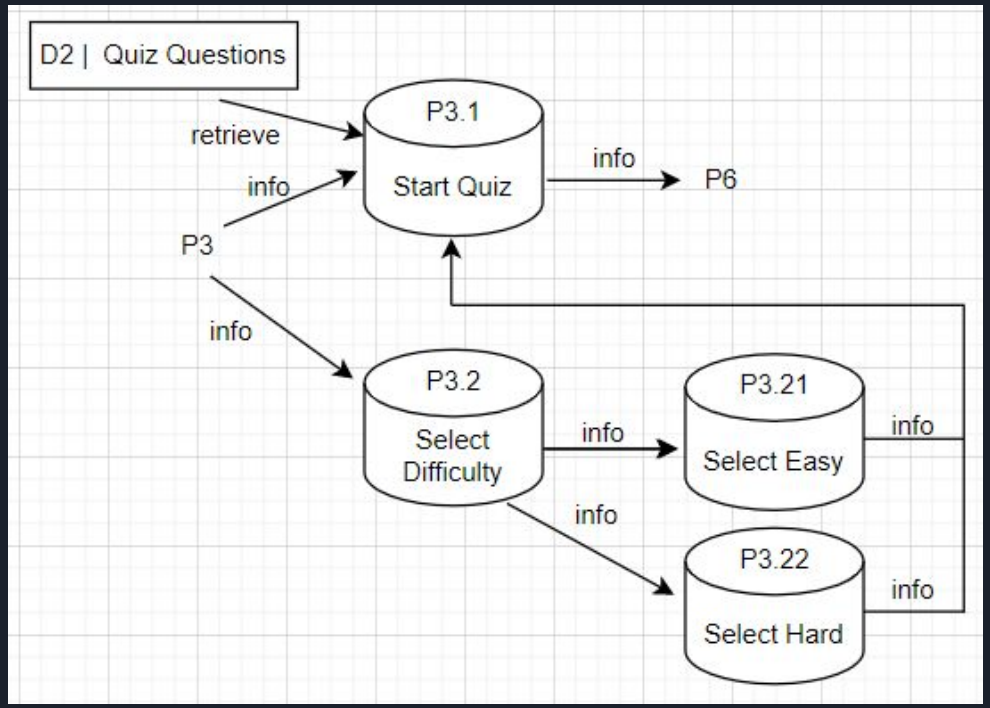

D2 | Quiz Questions
retrieve
D1 | User Accounts
retrieve
P3
Select New Quiz
info
P6
Match Facial Expressions and Emotions
info
P9
View Quiz Score
info
P4
Select Facial Detection
info
P7
Play Facial Expression Game
info
P1
Login from Start Menu
info
P2
View Main Menu
info
P5
Select Quit
info
P8
Closes Program

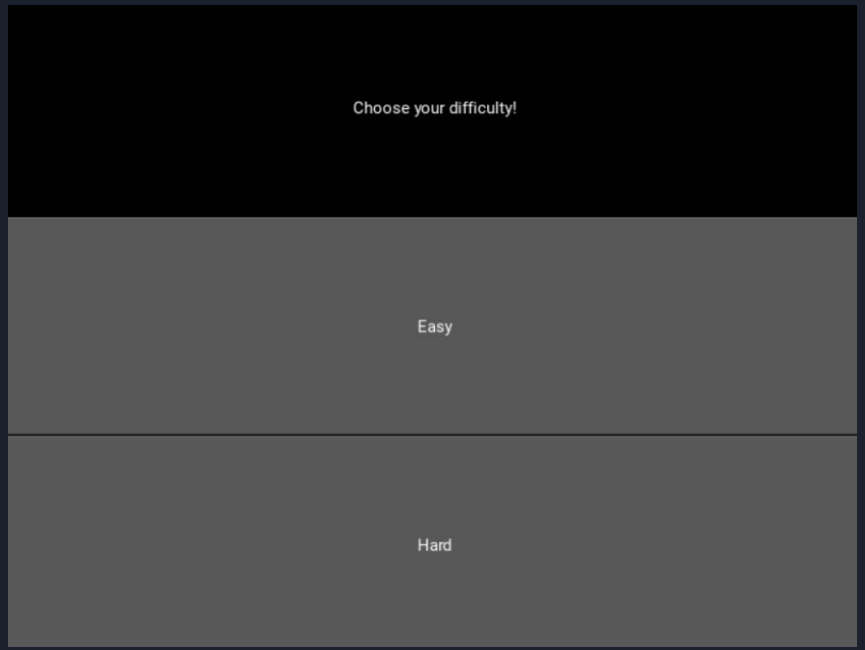
Child DFD - Create User Account/ Login





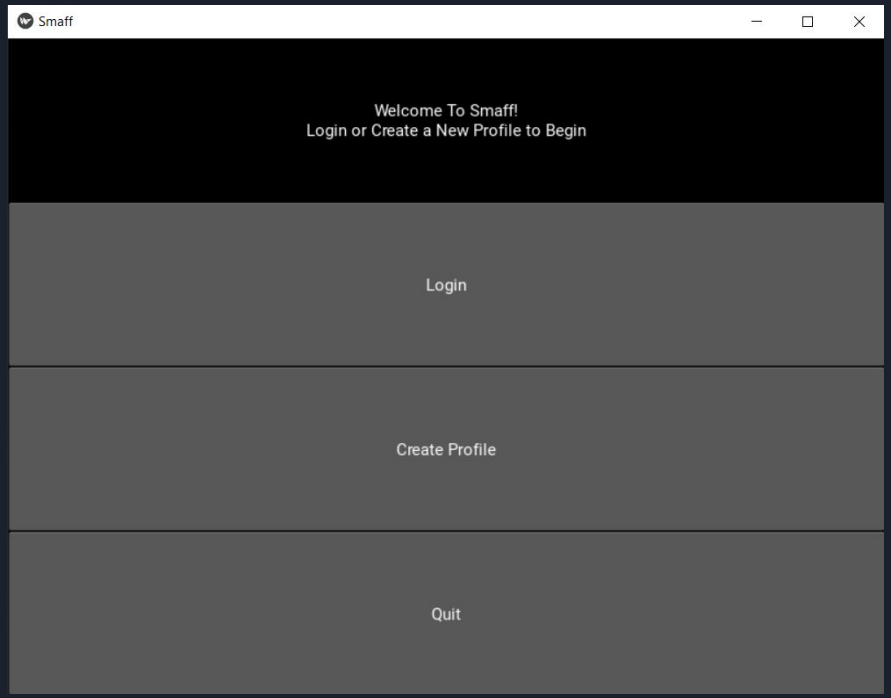
Child DFD - Quiz Selection





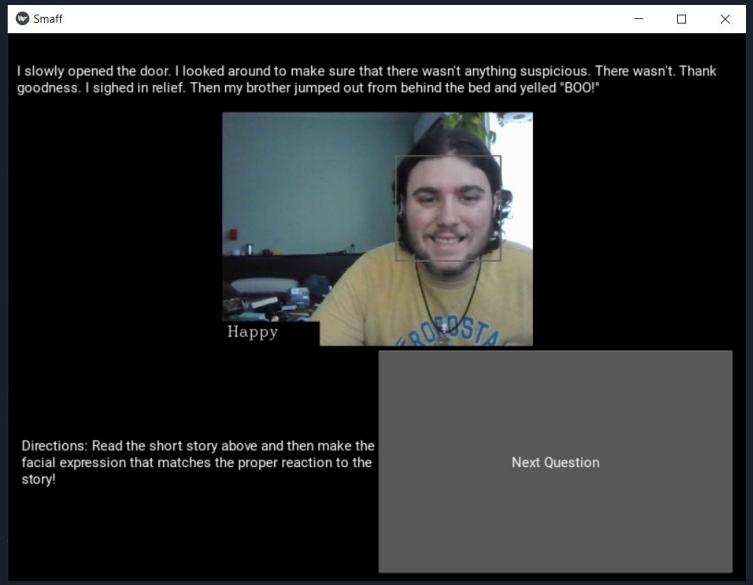
Implementation - Frontend

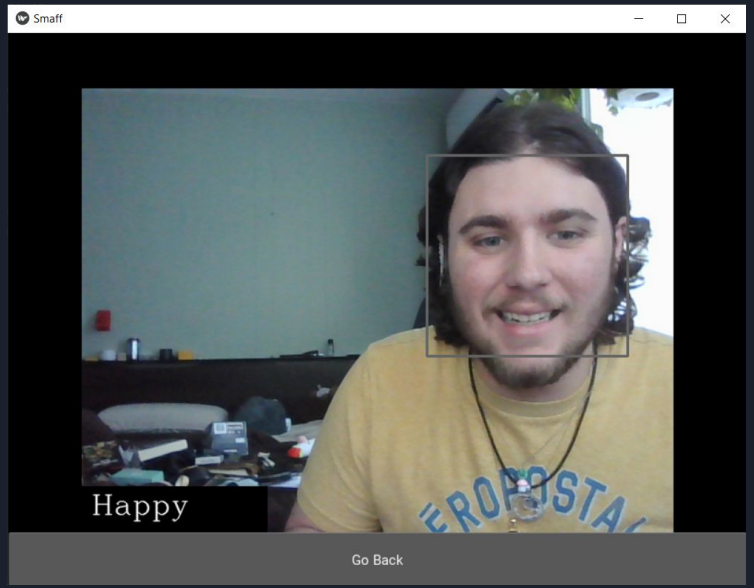
- The GUI was coded in Python using PyCharm and was implemented using Kivy.



OpenCV - For Display

- OpenCV was used to access the user's webcam in order to capture facial expressions during the quizzes





Implementation - Backend

- Machine Learning with Tensorflow and Keras
- Convolutional 2D Neural Network
  - Allows for filters and classification
  - Trained model with 4 emotions
    - Multiclass classification of Happy, Sad, Angry, Surprised
    - Training dataset supplied by Kaggle

```
model = tf.keras.models.Sequential()
model.add(keras.Input(shape=(200, 200, 3)))
model.add(tf.keras.layers.Conv2D(16, (3, 3), padding='same', activation='relu'))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.MaxPool2D(2, 2))

model.add(tf.keras.layers.Conv2D(32, (3, 3), padding='same', activation='relu'))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.MaxPool2D(2, 2))

model.add(tf.keras.layers.Conv2D(64, (3, 3), padding='same', activation='relu'))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.MaxPool2D(2, 2))
```

Implementation - Backend

- Multiclass classification
  - Categorical parameter for model.fit()
  - Softmax for dense layer
- Compile
  - lr of 0.0001
- Results
  - Accuracy of 99%

```
model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(512, activation='relu'))
model.add(tf.keras.layers.BatchNormalization())

model.add(tf.keras.layers.Dense(256, activation='relu'))
model.add(tf.keras.layers.BatchNormalization())

model.add(tf.keras.layers.Dense(120, activation='relu'))
model.add(tf.keras.layers.BatchNormalization())

model.add(tf.keras.layers.Dense(4, activation='softmax'))

opt = tf.keras.optimizers.Adam(lr=0.0001)
model.compile(loss="categorical_crossentropy", optimizer=opt, metrics=['accuracy'])

model.fit(train_dataset, batch_size=120, steps_per_epoch=200, epochs=15)
```

```
Epoch 10/15
200/200 [==============================] - 217s 1s/step - loss: 0.0814 - accuracy: 0.9893
Epoch 11/15
200/200 [==============================] - 197s 987ms/step - loss: 0.0784 - accuracy: 0.9886
Epoch 12/15
200/200 [==============================] - 189s 945ms/step - loss: 0.0720 - accuracy: 0.9884
Epoch 13/15
200/200 [==============================] - 203s 1s/step - loss: 0.0527 - accuracy: 0.9948
Epoch 14/15
200/200 [==============================] - 196s 982ms/step - loss: 0.0422 - accuracy: 0.9950
Epoch 15/15
200/200 [==============================] - 217s 1s/step - loss: 0.0404 - accuracy: 0.9956
```

OpenCV For Classification

- Stream webcam feed to OpenCV
- Use pre-packaged haar cascade classifier to identify faces
- Capture results of facial classifier as grayscale image
- Send image to our model and return prediction

```
def update(self, *args):
    ret, frame = self.feed.read()
    height, width = frame.shape[:2]
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    cv2.rectangle(frame, (0, height - 50), (200, height), (0, 0, 0), thickness=cv2.FILLED)
    faces = self.face.detectMultiScale(gray, minNeighbors=5, scaleFactor=1.1, minSize=(25, 25))

    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (100, 100, 100), 2)
        facex = frame[y:y + h, x:x + w]
        facex = cv2.cvtColor(facex, cv2.COLOR_BGR2GRAY)
        facex = cv2.resize(facex, (200, 200))
        facex = facex / 255
        facex = facex.reshape(200, 200, -1)
        facex = np.expand_dims(facex, axis=0)
        prepred_face = self.img_model.predict(facex)
        prediction = np.argmax(prepred_face, axis=1)
        lbl = None
        if prediction[0] == 0:
            lbl = 'Angry'
            self.ans = 'Angry'

        if prediction[0] == 1:
            lbl = 'Happy'
            self.ans = 'Happy'

        if prediction[0] == 2:
            lbl = 'Sad'
            self.ans = 'Sad'

        if prediction[0] == 3:
            lbl = 'Surprised'
            self.ans = 'Surprised'
```

Demo

Summary/Overview

- Create application that would help with facial recognition (specifically geared towards autistic children)
- Learn required software for programming application.
- Implement and test.
- Deploy.

Future Directions/Considerations

- User Interface
  - Different color palettes/more accessibility
- Additional features
  - More training for models
  - **More questions, situations**
  - Information/educational section about the different expressions?
  - More catered statistics
    - Overall performance/record
    - Which emotions they tend to get wrong
      - Utilize user statistics to query questions database
    - Improvements

Conclusion

- Challenges
  - Conflicting schedules
  - Time constraints
    - **Change of project topic**
      - Less time to work on project
  - Navigating new development tools/environments
  - Usage rights for facial expression dataset limitations
- Learning outcomes
  - Communication

Application of Knowledge

Dylan:

- Data science courses
- Software Engineering
- Personal and Work related projects

Jennifer:

- Data mining
- Big Data Analytics
- Experience from internships

Annie:

- Software engineering
- Program language concepts
- Internet programming

Loyal:

- Software Engineering
- Program language concepts

Team Member Contribution

- Dylan : Computer Vision, GUI
- Jennifer: Machine Learning
- Annie: Assets and research
- Loyal: Frontend

References

[1]]"MorphCast - Best Face and Emotion Recognition AI SDK | Face Recognition Javascript", *MorphCast*. [Online]. Available: https://www.morphcast.com/sdk/.

[2]S. Rucker, "Emotion Demo", *GitHub*. [Online]. Available: https://github.com/kairosinc/api-examples/blob/master/python-demo/static/docs/emotion/Emotion.md.

[3]"Micro Expressions Training Tools", *Paul Ekman Group*. [Online]. Available: https://www.paulekman.com/micro-expressions-training-tools/.

[4]"Face Recognition Features from Kairos", *Kairos*. [Online]. Available: https://www.kairos.com/features.

[5]"Autism Spectrum Disorder", *National Institute of Mental Health (NIMH)*. [Online]. Available: https://www.nimh.nih.gov/health/topics/autism-spectrum-disorders-asd.

[6] Maenner MJ, Shaw KA, Bakian AV, et al. *Prevalence and Characteristics of Autism Spectrum Disorder Among Children Aged 8 Years — Autism and Developmental Disabilities Monitoring Network, 11 Sites, United States, 2018. MMWR Surveill Summ 2021;70(No. SS-11):1–16. DOI: http://dx.doi.org/10.15585/mmwr.ss7011a1*

[7] https://www.kaggle.com/datasets/chiragsoni/ferdata