Big Classroom Support

Team Name: sddec20-27

Client: Prof. Lotfi Ben Othmane

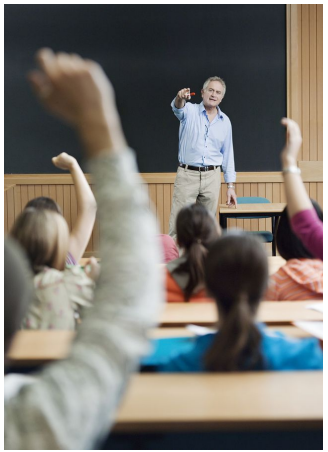
Advisor: Prof. Lotfi Ben Othmane

URL: http://sddec20-27.sd.ece.iastate.edu/

Brendan Niroula, Zechen Huang, Jian Kai, Ali Al Ahbabi

Problem Statement

Many students prefer to be called by their names when they interact with their instructors as this helps the students stay engaged. Referring to each student by name becomes difficult for the instructor when in large lectures. This project aims to develop an assistant to help instructors identify a student's name instantly without having to memorize them.



Goal

To deliver an application with Smart Glasses which will display a student's name simply by looking in their direction.

Functional Requirements

- Students and Instructors can sign in
- Students can upload their photo in the app
- Instructors use smart glasses to live stream to remote server
- Remote server can find the students' name
- Instructor sees students' names via the mobile app

Non-functional Requirements

- App is secure (user data is secure)
- Processing an image takes less than 1 second
- Can handle 100 concurrent users
- Can store 400 units of picture data in the database

Market survey

- Face 6:
  - Attendance software using facial recognition
  - Cost: 6 digits
  - On drones and security cameras
  - www.face-six.com

- Google glass:
  - Doesn’t have official facial recognition software
  - Product was discontinued in 2015
  - Cost :$1500 explorer version
  - Current price since google no longer sell it

Resource/Cost Estimate

Resource:

Physical Resource - Smartglass - Vuzix Blade, Android phone

Software Resource - Android Studio, Python IDE, Github, Django

Cost: Vuzix Blade - 700 USD

Potential Risks

- Smart glasses Cost
  - $700
- Huge learning curve:
  - Python using Django
  - RTMP protocols
  - facial algorithms
  - Vuzix Smart Glasses
- Accuracy issues:
  - The algorithm we use for face recognition is not 100% accurate.

VUZIX®
View the Future®

Mitigation

- Costs
  - Mitigate costs by developing a versatile app which can be used on any smart glasses.
- Learning Curve
  - Pair programming and extensive research.
- Facial Recognition Accuracy
  - Adjusting the scale and adding more data points to better the accuracy.

Technical/Other Constraints/Considerations

Economic/Market Requirement:

- Budget = $700
- Components = Smart Glasses with live streaming ability

Environmental Requirements:

- Sufficient lighting in the room for video capture
- 50 students maximum in each classroom
- Clear line of sight to each student

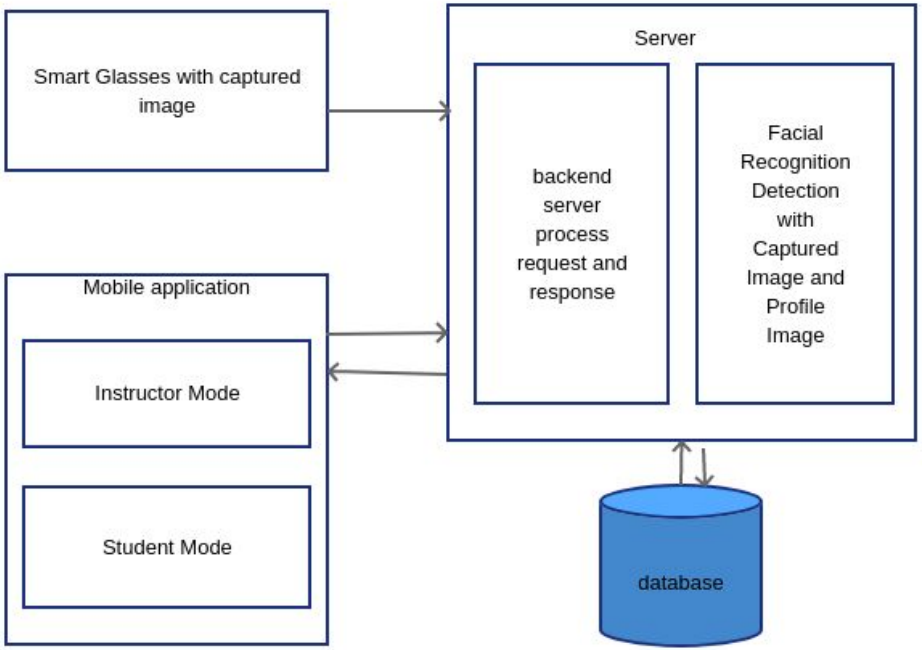
System Design

Functional Decomposition

This project has three components:

- Smart Glasses Android Application
- Mobile Android Application
- Backend Django server with Facial Recognition implemented

Conceptual Sketch For Solution



Detailed Design

HW/SW/Technology Platform(s) used

- Hardware: Vuzix Blade Smart glasses that is running Android system 5.1.1
- Software:
  - Backend : Python (Django RESTful API)
  - Frontend : Java - Android studio for mobile and smartglass apps
  - Open Source Face recognition algorithm provided by ageitgey on Github and OpenCV
  - Database: SQLite - Django generated default database using serializer

Server

- Django RESTful and face recognition server- handle request and response, run face recognition to identity student in captured image
  - Django channels - websocket to send the students name.
  - Face Recognition - Using face recognition algorithm from github to detect and encode people in uploaded pictures. Store encoded people in database with name for comparing. When comparing, find people in picture, encode the target people, and comparing to every encoded people in database to find out people in target picture.
- RTMP server - set up live stream url to receive and stream live video
  - Configured RTMP NGINX server

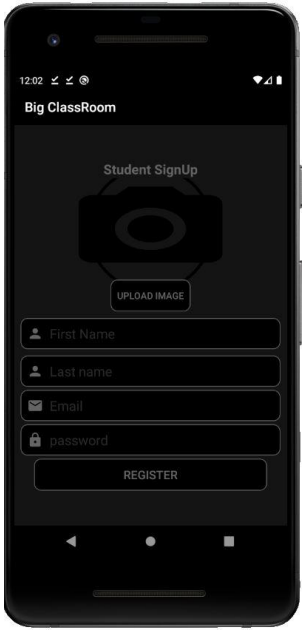
Smart Glasses

- Development was done entirely on Android Studio, components include:
  - Broadcasting the stream to the server using RTMP protocol.
  - WebSocket connection with the client who enters a unique client code.
  - Sending images to the server for face recognition processing using HTTP Requests.
  - Receiving the name from the server of the student within the image using HTTP requests.

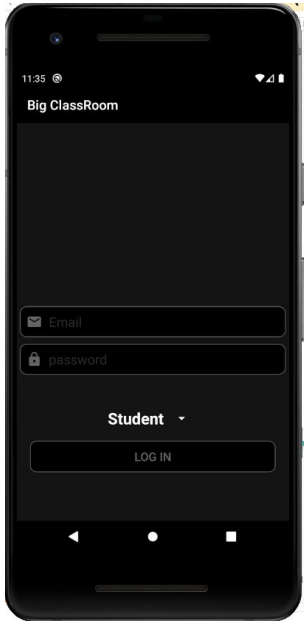
Mobile Application

- The mobile application is developed using Android Studio.
  - Java
  - Login authentication is done through HTTP protocols
  - Receiving live stream from the smart glasses is done through RTMP protocol
  - Sending the request of taking a picture to the smart glasses is done with WebSockets
  - Displaying the returned name is done through WebSockets

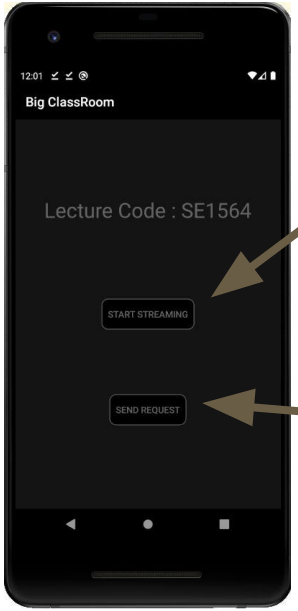
UI Design



Student sign up screen



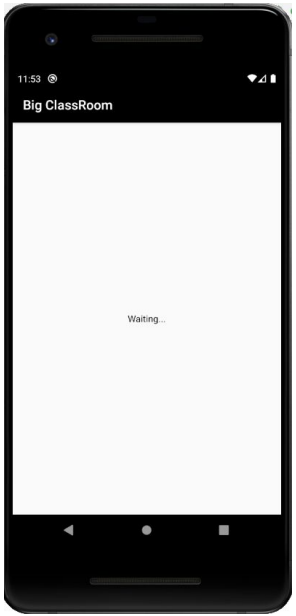
Sign in screen



Receives RTMP stream

Send a request to the smart glasses to take a picture

Instructor screen

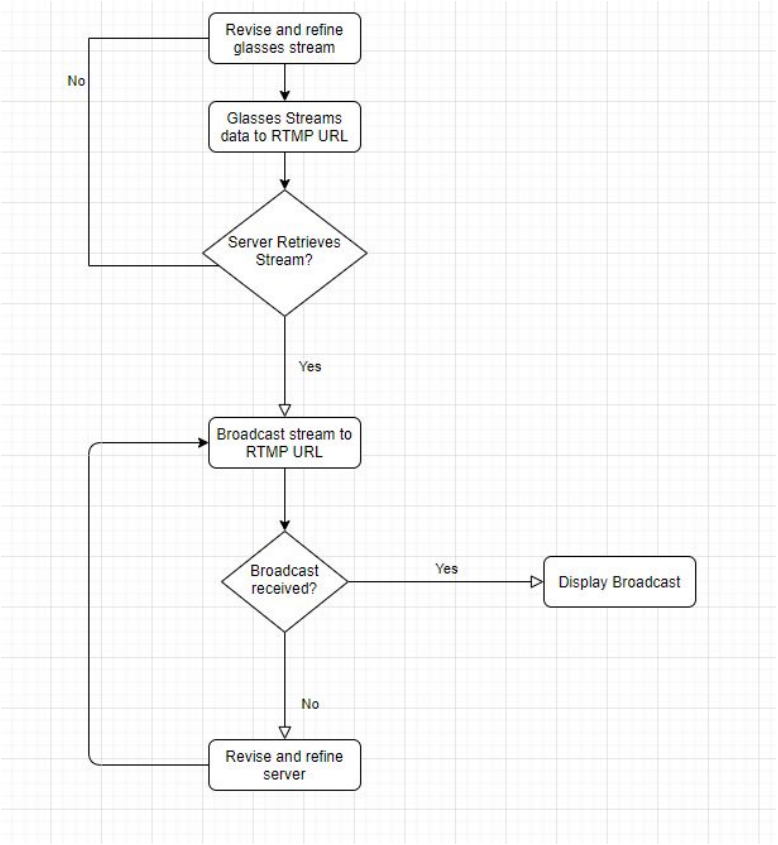


Display returned name

Testing

Test Plan

As we have broken up the project into 3 separate components, our plan was to test these individually first. For this we will use tools such as JUnit, OBS, VLC and postman. Once we completed individual testing we moved onto testing the integrated components as seen in the diagram.



Test Results

- Frontend Tests
  - Assertion Tests Passed
  - UI Tests Passed
- Postman Tests
  - API Tests Passed
- OBS & VLC
  - Stream is displayed successfully

Demo

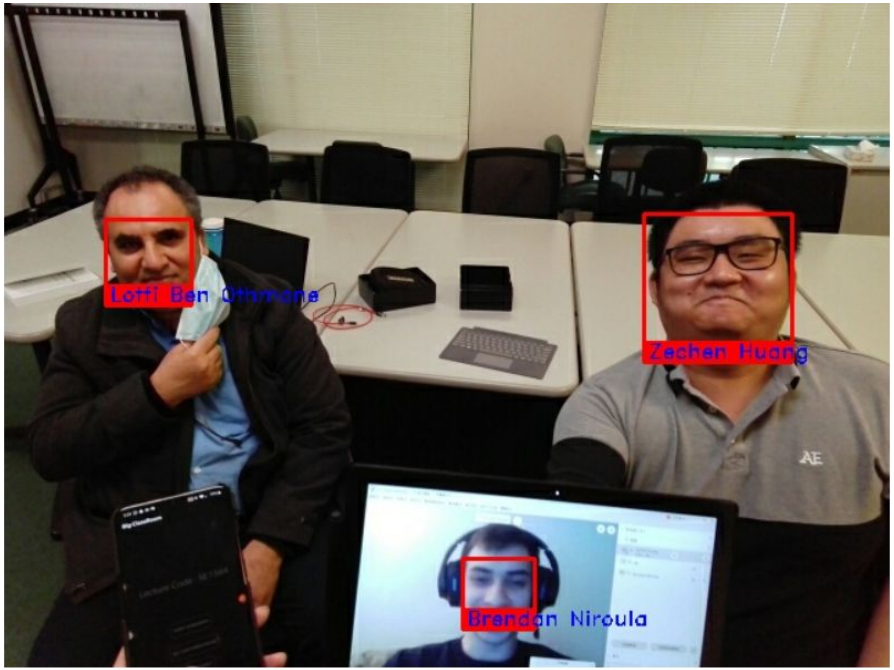
Mobile Application Demo

Live Stream Demo

Facial Recognition Demo

Field Testing & Client Feedback

10/31/2020 - we captured the image using vuzix blade smart glass with our application. Our client was happy with our progress and work we have put into this project.



Conclusion

Task responsibility/contributions of each project member

- Brendan Niroula -Smart Glasses Developer and Team Leader
- Ali Al Ahbabi - Front app developer with Android studio
- Zechen Huang - Face recognition server developer to set up server for image, opencv and face recognition
- Jian Kai Lee - Web server maintainer and developer to setup django framework and nginx for RTMP

Final Remarks

This project began with a large learning curve and was very daunting. As the year went on, it became a very fun and engaging project.

Lessons Learned:

- Virtual Communication
- Development Processes
- Smart glasses development
- Django development
- Opencv development
- Face-recognition development

Future Work

- Implement chat feature in mobile application
- Improve face recognition accuracy
- Displaying the returned image with the name on the mobile application

Questions?