Big Classroom Support for Instructors

02.23.2020

sddec 20-27

Ben Othmane, Lotfi - Client & Adviser

Team Members:
Brendan Niroula - Leader/ Smart Glass Developer
Zechen Huang - Facial Recognition Developer
Ali Al Ahbabi - Mobile Developer
Jian Kai Lee - Mobile Developer

Team Email:
sddec20-27@iastate.edu

Team Website:
ssdec20-27.sd.ece.iastate.edu

Executive Summary

Development Standards & Practices Used

- Hardware: Smart glasses that has live streaming feature
- Software:
    a. Backend : Python (Django)
    b. Frontend : Java - Android studio
    c. Face recognition algorithm
    d. Django default database using Python
- Engineering standards:
    a. Design Documents:
        - Use Cases
        - Activity Diagram
        - Component Diagram
        - Communication Diagram
    b. Documentation of code:
        - In line comments
        - JS Doc

Summary of Requirements

- Functional requirements
    - Students and Teachers can sign in
    - Students can upload their photo in the app
    - Students can send questions in chat room to teacher
    - Teacher uses smart glasses to live stream to remote server
    - Teacher sees students names and pictures via the app
    - Remote server can find the students' name

- Non-Functional requirements
  - App 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

Applicable Courses from Iowa State University

- COMS 227 Object-oriented Programming
- COMS 228 Intro to Data Structure
- COMS 309 Software Development Practices
- COMS 472 Principle of Artificial Intelligence
- SE 319 Construction of User Interface
- SE 329 Software Project Management

New Skills/Knowledge acquired that was not taught in courses

- Smart Glasses development
- RTMP protocols
- Python language
- Face recognition Algorithm

Table of Contents

1.   Introduction

1.1   Acknowledgement

Our team would like to express my special thanks to our adviser and client (Professor Ben Othamane, Lotfi) who gave us the chance to work on Smart Glass technologies in Big Classroom Supports for Instructors and assist us in developing and researching smart glass applications.

1.2   Problem and Project Statement

Professors everywhere are always tasked with memorizing students' names, this can be extremely challenging especially with large classrooms. As a result, we have taken upon us the task of making this easier for Professors using the power of technology. Our task is to develop an app which will allow professors to know students' names without having to memorize all of them. We will do this with the help of smart glasses. Our goal is to create an android and IOs app, which will display the students which the professor sees with their glasses onto their phone along with their names next to them.

1.3   Operational Environment

Big classroom project contains two main parts. First part is the smart glasses which will be worn by the instructor inside the classes. So there should be sufficient lighting in the room in order for the camera of the smart glasses to capture a clean video. The second part is the mobile app which will be running on all new generation phones with iOS and Android operating systems.

1.4   Requirements

Functional Requirement:

- Students and Teachers can sign in
- Students can upload their photo in the app
- Students can send questions in chat room to teacher
- Teacher uses smart glasses to live stream to remote server
- Teacher sees students names and pictures via the app
- Remote server can find the students' name

Non - functional Requirement

- App 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

Economic/Market Requirement:

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

Environmental Requirements:

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

UI requirement:

- Question lists
- Sign in/ Sign up
- Profile tab
- Upload photo tab

1.5   Intended Users and Uses

There are two types of users: instructors and students. Instructors are going to wear smart glasses during the lecture. The smart glasses will stream live video to a mobile app which will be installed on the instructor phones. The instructor will be able to sign in for a teacher mode in the app. This mode allows instructors to read students' questions along with their names. The instructor also will have the option to capture an image or a short video of the live stream. Students will be able to sign up into students mode which will ask them to upload their photos. Also, they are going to use the mobile app to ask the instructor questions anonymously.

1.6   Assumptions and Limitations

Assumptions:

- The maximum number of students in a class is 60 - 70.
- The maximum number of names which will appear is 5.
- The app will only be used in the U.S. for universities.
- The maximum number of students in the database will be 300.

Limitations:

- The glasses must be lightweight and not distracting.
- The project must be finished by December 2020.
- The budget must not exceed $700.
- The app must be accessible to all people with disabilities.

1.7   Expected End Product and Deliverables

The end product of this project is an android/IOs app which has capability to receive live stream data from a pair of camera glasses and display the names of students onto the phone screen using face recognition. This app will also allow students to communicate with the professor during class via the app. The expected delivery date for this product is December 2020.

2.   Specifications and Analysis

2.1   Proposed Approach

After we have the pictures from students, and analyze them with algorithms, we will have students' data stored in a server for future use (Functional requirements: Students and Teachers can sign in; Students can upload their photo in the app.)

We use smart glasses to capture pictures or video streaming to the server, then the server will identify the students and the one who speaks. After that, the server will send the name and the picture to the phone app (Functional requirements: Teacher uses smart glasses to live stream to remote server; Teacher sees students' names and pictures via the app; Remote server can find the students' name.)

We also need a discussion room within the app where students can send questions. The teacher can see these questions through the phone app (Functional requirements: Students can send questions in chat rooms to the teacher.)

Also, the process of this approach needs to be finished in no longer than 1 second, and at least 100 users should be supported. The server should be able to store at least 400 students' data. The data flow between devices should be secured (App 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.)

We have found decent smart glasses named VUZIX that can take pictures or video. We also have found several face recognition algorithms that can be used in our project, and they are written by Python or Nodejs. We have tested several api that are provided by these algorithms like getting people's face data from the pictures.

The standard we use inside smart glasses is Android Lollipop, and Android app for the phones. For the server, we use Django and Face recognition algorithms with Python.

2.2   Design Analysis

We have found decent smart glasses named VUZIX that can take pictures or video . We also have found several face recognition algorithms that can be used in our project, and they are written by Python or Nodejs. We have tested several api that are provided by these algorithms like getting people's face data from the pictures.

So far, the algorithm works fine as we followed the instructions from github.

In the future, we will use the embedded Android system inside glasses. We have observed that finding the right glasses is very hard, and we will stay on this one. Our idea is using Server-client architecture. The Android system is a strength because it has a mature development kit to use. Weakness may be the system may not be efficient enough while in use because glasses are too small to put powerful components.

2.3   Development Process

We are going to use Agile because it is one of the most used development processes in the world. We are going to use server-client architecture. We are going to use gitlab to manage the tasks we need to do. We are going to use Slack for team communication.
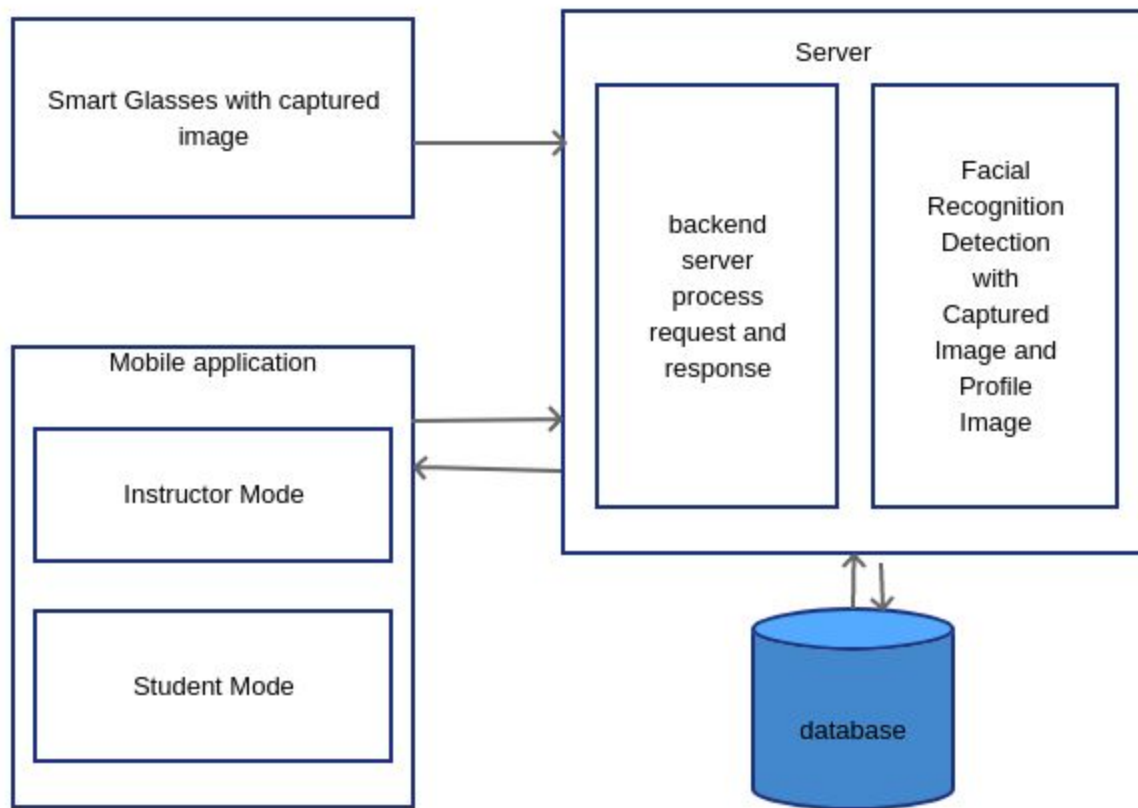
2.4   Conceptual Sketch

Figure 1

3. Statement of Work

3.1 Previous Work And Literature

There is an ongoing augmentation of smart glasses with reality specs in smart glasses. As facial recognition technology becomes significant in the public domain, it is essential to explore how it can be used in the education sector. The technology will come in to solve cases of campus security, detection of the students' movement, and automatic registration. The technologies can add to the routine systems of monitoring and surveillance, which is a common culture in schools.   Major tech companies are making strides in creating AR smart glasses that may be the future of artificial intelligence. Apple, Facebook, and Amazon are making significant steps in that direction.

One application of facial recognition in schools is in monitoring attendance. Routinely, teachers have to take roll-calls for a large number of students, which is cumbersome. In Australia, the education system uses "LoopLearn" technology for roll calls, which saves time in the classroom (Andrejevic and Selwyn 3). The technology helps to save up to 2 hours as teachers take students to roll calls using the cameras. Additionally, there is a growing

interest for online learners to use facial detection as they engage in online classes. Online courses have, for some time, suffered from integrity issues as students miss classes or send others to represent them (Valera, Valera, and Gelogo 17). The access to educational content has to be limited to the learners of the specific courses. Thus, technology developers are working closely to come up with AR technologies that can address this trend.

The use of Google Glass in schools has helped learners and teachers alike to show the information they gather during their natural interaction with nature. Technology has many possibilities and can allow learners to gain experience in various fields. It can also be used under improved conditions to document information that students learn in the field, which can help in monitoring class attendances, especially for field trips. It is apparent that facial recognition will be useful in the school environment as educators monitor the learning process, security, and the integrity of the pedagogical processes.

Andrejevic, Mark, and Neil Selwyn. "Facial recognition technology in schools: critical questions and concerns." *Learning, Media and Technology* (2019): 1-14.

Valera, Jasmine, Jacinto Valera, and Yvette Gelogo. "A review on facial recognition for online learning authentication." *2015 8th International Conference on Bio-Science and Bio-Technology (BSBT)*. IEEE, 2015.

3.2 Technology Considerations

The current technology proposal will look at the possibility of having specific technology that teachers can use to identify the students they teach online. The technology aims at integrating the facial recognition software in the webcams of students attending online classes in platforms such as Moodle.

The technology can easily identify the learners who access learning material from the university or college database accurately. It will allow tutors to know which students attend classes regularly in the virtual environment. It will also facilitate the commitment of learning and avoid impersonation during online classes and examinations.

The technology may suffer from hacks from intruders who can compromise the integrity of the whole system. Cyber-criminals may ensure that they alter the devices to suit their needs, thus beating the logic behind the security aspects. More so, the idea will be costly to implement, thus requiring high capital input that may be inaccessible to many.
The institutions and service providers who intend to have the technology in their schools will have to pay the price. The technology will also be subsidized as it launches to facilitate quick absorption. The technology will also have a security feature that can limit its

vulnerability to hacking and other security issues associated with AR. Collaboration with manufacturers to have cheaper alternatives will be prudent to ensure that the technology exists in a variety of ranges that everyone can afford.

3.3 Task Decomposition

Big Classroom support for instructors

1. Project Analysis
   a. Discussions with client
   b. Prepare project plan
   c. Client approval
   d. Project kickoff
2. Big classroom product
   a. Hardware
      i. Gather information about smart glasses in market
      ii. Gather smart glasses requirements
      iii. Choose smart glasses that satisfy the requirements
      iv. Advisor approval of the smart glasses
      v. Order smart glasses
   b. Software
      i. Gather Software requirements
      ii. Start developing Front-end that's satisfy the requirements
         ● Login page
         ● Uploading student pictures and send it to server
         ● Receive stream from smart glasses
         ● Instructor mode - allow instructor to take a picture from the stream and send to the server
         ● Chat feature for students
      iii. Start developing Back-end that's satisfy the requirements
         ● Login authentication
         ● Store picture with student info in database
         ● Send picture to facial recognition software
         ● Receive names from facial recognition software
         ● Send students name to front-end
      iv. Start developing Facial recognition software
         ● Receive picture from back-end
         ● Compare faces in the picture with pictures in database
         ● Return student names to backend
      v. Start developing smart glasses software
         ● Send live stream to front-end
      vi. Testing
         ● Test connection between all parts

- Test software functionality
- Fix bugs and update softwares
  c. Launch product

3.4 Possible Risks And Risk Management

Cost:

The cost of smart glasses is not cheap and by checking the price, most of them are higher than our budget.

Solution: Our advisor gave us more money, and we found a sufficient one with the lowest cost.

Knowledge of area: We plan to use Python and Django as our main backend language, but we are not so experienced with it.

Risk management: We are trying to learn it and use it with our best effort, and it looks good.

Equipment: Same as cost, and programming on smart glasses is also a challenge for us.

Risk management: We are trying to learn it with our best effort.

Accuracy issues: The algorithm we use for face recognition is not 100% accurate, and needs to adjust by time.

Risk management: When finishing the part of using the algorithm, I need to adjust it to make sure it reaches the best result for us.

3.5 Project Proposed Milestones and Evaluation Criteria

We have split our task into four parts: an android phone app, smart glasses app, backend framework, face recognition and database.

1. We have the smart glasses we need, which is the only equipment we need. Also, we decided which platform we are going to use.
2. The first milestone should be the four parts we mentioned to run some simple functions and all environments(Python, Django, Android app on phone, Android app

on smart glasses, and the face-recognition environment) should be set up. We will use postman and Mock unit tests to test each part separately.

3. The second milestone should be every part to be connected and decide what kind of data flow and encryption type for this project. We will still use postman and Mock unit tests to test, and the physical code to test each other.
4. The third one should be the four parts to be finished, and successfully demonstrated. The test should be the real project test.
5. The Fourth one should be based on demonstration, adjusting the details of the project, and doing as much testing as we can, to polish the whole project. The test should be done real world responds.
6. The Fifth milestone should be adding some new features, and only push it when it is working with a current project without conflicts. Testing should be also using postman and mock unit tests.

3.6 Project Tracking Procedures

We are using gitlab which has completely tracking processes called milestones, issues, and snippets. Those tools can help us manage the whole project and also assign each other tasks convenient.

Also, we use whatsapp and slack to communicate and discuss.

3.7 Expected Results and Validation

The desired outcome of this endeavor is to create a product which supports instructors in their day to day lives by easing their troubles of memorizing many students' names. Using the advanced technology we have today including facial recognition, smart glasses and smart phones, our product should be user friendly and very useful to our users. We would like this product to be approved and endorsed by instructors around the world.

At a high level, we will be satisfied with our product when instructors and students alike, both approve and endorse our product as intuitive, user friendly, and useful. Although we will be able to confirm the outcome and approval, we will continuously maintain the product so that these standards are constantly met.

4. Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

	Feb 20	Mar 20	Apr 20	May 20	Aug 20	Sep 20	Oct 20	Nov 20	Dec 20
Project Analysis: discussions with client and prepare project plan	▓								
- Client approval and Finalize project plan		▓							
Research and discussion Smart Glass in the market and gather Requirements	▓								
Start Develop Front-end Interface in Android: Login page, upload picture function		▓	▓	▓					
- Develop Instructor Mode, Chat features and receive stream from Smart Glass		▓	▓	▓					
Develop and Learn to create Smart Glass App in Android			▓	▓					
- implement data transfer Smart Glass and Mobile				▓	▓				
- live streaming function				▓	▓				
Setup Back-end Django Restful API framework and implement Login Authentication				▓					
- Handle sending picture and information between Server and Android					▓	▓	▓		
- Implement Facial Recognition Analysis Software					▓	▓			
Develop the Facial recognition Analysis software in Server						▓	▓		
- Implement open source Facial Recognition Algorithm						▓	▓		
- Compare and testing the Facial Recognition Results						▓	▓		
write unit, GUI and Integration tests to End to End System							▓	▓	
Run the usability of end to end system							▓	▓	
Final bugs fix and improvement									▓

Figure 2

The initial stage of the project was to project analysis including discussion with client and prepare the project plan and design to the client approval. It takes one month to complete the task. Next, we start to research the similar product in the market and evaluate the similarity with existing products in the market and our project. In the meantime, we research the smart glass to find the best suitable smart glass that fits our project scope. It will be done by the end of March 2020.

The second stage involves android development with implementation of login, upload picture function, instructor mode and chat feature. It will take place between March 2020 to May 2020. The Smart Glass Development will start from April 2020 to August 2020 including setup data transfer from smart glass to server and live streaming. In August 2020, we will start implementing more features to our back end server that can handle requests from smart glass and mobile. It will take 3 months to fully implement. Next, we will implement the open source facial Recognition Algorithm in server and testing the accuracy of facial recognition. It will be done by October 2020.

The third stage includes writing unit, GUI and integration test for the whole system run the usability testing of whole systems. It will start in October and end in November 2020.

The Final Stage will take place in December 2020 to do final improvement and fix bugs.

4.2 Feasibility Assessment

This project will provide the Instructor to identify student names and information in a convenient way by using the smart glass and mobile application. The project will have the instructor wearing the smart glass in the class and take a picture of the students. The mobile application will display the student's information based on facial recognition analysis and instructors are able to call the student name without memorizing all the

student names. The foreseen challenges we will encounter in this project is the learning curve in smart glasses development and the accuracy of the facial recognition prediction.

4.3 Personnel Effort Requirements

The Table below shows the estimated hours needed in each area. It breaks down into six categories: Project Analysis, Research and requirement, Front-end Development, Smart Glass Development, Back-end Development, Facial Recognition Software and testing.

Project Analysis	**15 hours**
discussions with client prepare project plan	10 hours
Client approval and Finalize project plan	5 hours
Research and Requirement	**30 hours**
Research and discussion Smart Glass in the market and Requirements	30 hours
Front End Development	**90 hours**
Login page, upload picture function	25 hours
Develop Instructor Mode, Chat features and receive stream from Smart Glass	65 hours
Smart Glass Development	**160 hours**
Develop and Learn to create Smart Glass App in Android	30 hours
implement data transfer Smart Glass and Mobile	50 hours
live streaming function	80 hours
Back-end Development	**110 hours**
Handle sending picture and information between Server and Android	50 hours
Implement Facial Recognition Analysis Software	60 hours
Facial Recognition Software	**120 hours**
Implement open source Facial Recognition Algorithm	60 hours
Compare and testing the Facial Recognition Results	60 hours

Testing	100 hours
write unit, GUI and Integration tests to End to End System	25 hours
Run the usability of end to end system	25 hours
Final bugs fix and improvement	50 hours

Figure 3

4.4 Other Resource Requirements

Required physical resources for this project are the smart glasses, an android phone, a computer for development. Software resources required are android studio and python IDE.

4.5 Financial Requirements

The only financial requirement for this project is the smart glasses which cost $700.

5. Testing and Implementation

5.1   Interface Specifications

Define and describe any hardware or software interfacing that is being used for testing aspects of the project.

The software interface of the testing aspects of our project consist of

1. Define the type of testing modules needed in our project
2. Define the type of testing in individual components
3. Design and discuss the testing method
4. Determine the expected output from the testing
5. Implement the testing to components and compare actual output and expected output
6. Revise if needed

For the glasses application, we are using JUnit testing within Android Studio to test critical functions within the application using negative and positive testing.Similarly with our

mobile application, we implement JUnit testing. In the server side, in order to test the http url request and response, the postman allows us to compare the outputs from the server.

5.2   Hardware and software

JUnit testing will allow us to test our functions individually to ensure each component is working as designed.

For hardware, the vuzix smart glass will be used to capture the video and test for the smart glass application. Android phones will be needed for installing mobile applications and test mobile application functions.

For software, we will use Android provided testing library to perform the unit testing and integration testing of mobile applications. Django testing library is needed for django server. We use Espresso to simulate real user behavior on mobile applications and postman to test the HTTP request and response. In terms of live streaming testing, we use open broadcast software to broadcast to the RTMP url and listen to the url in VLC media. VLC provide live stream

5.3   Functional Testing

For functional testing of our project, we will write Junit testing for our mobile application and smart glass since the smart glass is running android. We include django unit testing in our server. Also, the front end app is going to be tested using test cases since it now has the login feature and displaying the live stream. Our system testing includes running all applications and devices correctly.

Our client and professor will perform acceptance testing by using the mobile application and smart glass at weekly meetings to demonstrate our current progress and functionality to meet the requirements.

5.4   Non-Functional Testing

For the non-functional testing, our project includes performance and stability of live streaming video, mobile usability and data transfer security.

Non functional tests include:

1. Stream delay
2. Audio
3. Security of data transfer
4. Mobile application usability
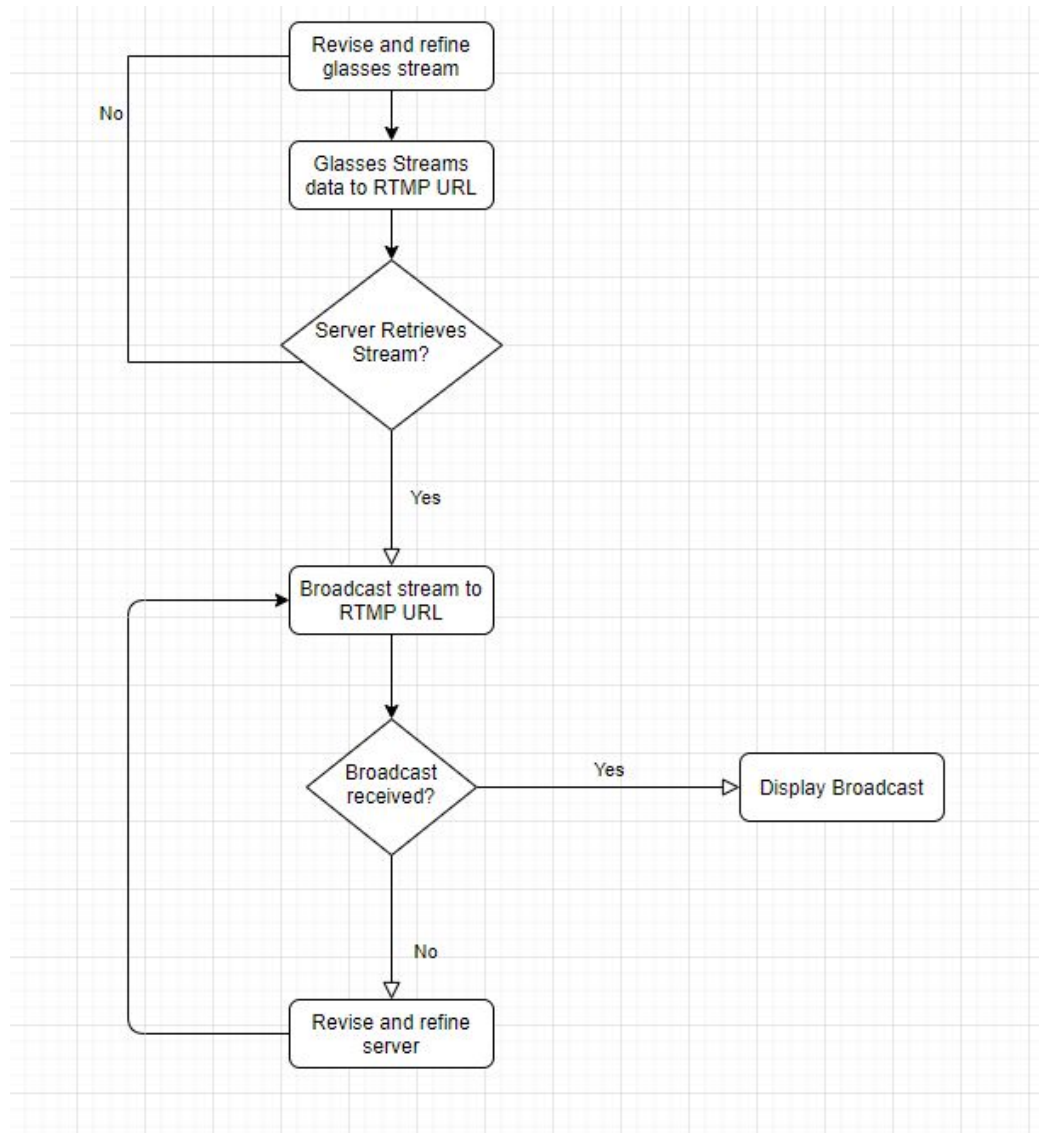
5.5 Process

Figure 4

 To test the glasses application, JUnit tests were put in place to verify responses of components within the code.

The front end app login feature is tested by a set of different inputs to verify that the feature is working and communicating with the backend successfully.

5.6   Results

The JUnit testing for the RTMP streaming on the glasses side all came back as expected. As we progress with this project, the tests will become more comprehensive. Also, based on the test cases that we executed for testing the mobile app it all went through successfully. All inputted data showed up in the database and we were able to retrieve them back to verify the sign in operation. Those successful test cases indicate that the mobile app has the desired connection to the backend. The other feature of the mobile app is displaying the live stream feature. The result of testing that feature shows that our mobile app has a stable connection with the backend since it plays the live stream for both video and audio without problem.

Some implementation issues we ran into was developing the camera application for the glasses. In order to do this from scratch it would require extensive knowledge of encoding, protocols, and the android camera sdk. This is out of the timeline and scope of the project, however we were able to find open source code which assisted us with this task and expedited the process.

6. Closing Material

6.1 Conclusion

To summarize the work we have done thus far, we have created a front end application to log in and display streaming data. We have created a server which can distribute the stream to wherever it needs to go. We have also used open source software to connect this all together using smart glasses. We also have another server running face_recognition process, but has not been connected yet.We are currently working on processing the streaming data through facial recognition software. Our goals are to have a well made application which professors can use to help them in their day to day lives. The best course of action to achieving this is constant testing, constant revising and managing and getting feedback from professors. This is the best course of action as getting real feedback from our clientele will make our app better from the start.

6.2 References

Christie, Tom. "Django REST Framework." *Home - Django REST Framework*,
www.django-rest-framework.org/.

Dodgepong. "How to Set up Your Own Private RTMP Server Using Nginx." *OBS Forums*, 20
June 2014,
obsproject.com/forum/resources/how-to-set-up-your-own-private-rtmp-server-usin
g-nginx.50/.

Ant-Media. "Ant-Media/LiveVideoBroadcaster." *GitHub*, 22 Aug. 2019,
github.com/ant-media/LiveVideoBroadcaster.

Ageitgey. "Ageitgey/face_recognition." *GitHub*, 20 Feb. 2020,
github.com/ageitgey/face_recognition.

6.3 Appendices

I. Supporting Articles
    A. Ruether, Traci. "RTMP Streaming: The Real Time Messaging Protocol
        Explained." *Wowza Media Systems*, 25 Feb. 2020,
        www.wowza.com/blog/rtmp-streaming-real-time-messaging-protocol.
        -   gives a good explanation of RTMP and why to use it.

    B. Vuzix. "Vuzix Blade See-Through AR Smart Glasses Powered by
        Industry-Leading Waveguide Optics." *Vuzix Blade® | Augment Reality (AR)
        Smart Glasses for the Consumer*,
        www.vuzix.com/products/blade-smart-glasses.

        - Vuzix Blade smart glasses information from the manufacturer.

    C. https://developer.android.com/reference/android/hardware/camera2/packa
        ge-summary - The SDK information for developing and using the android
        camera. The Vuzix smart glasses run on Lollipop 5.0 so you can program
        them just as a normal Android phone.

    D. Ant-Media. "Ant-Media/LiveVideoBroadcaster." *GitHub*, 22 Aug. 2019,
        github.com/ant-media/LiveVideoBroadcaster.

        - AntMedia provides an open source RTMP client which we can use for the
        Smart Glasses to stream.

E. "face_recognition Package." *face_recognition Package - Face Recognition 1.2.3 Documentation*, face-recognition.readthedocs.io/en/latest/face_recognition.html. - The API for the face recognition process.

F. Andrejevic, Mark, and Neil Selwyn. "Facial recognition technology in schools: critical questions and concerns." *Learning, Media and Technology* (2019): 1-14.

G. Valera, Jasmine, Jacinto Valera, and Yvette Gelogo. "A review on facial recognition for online learning authentication." *2015 8th International Conference on Bio-Science and Bio-Technology (BSBT)*. IEEE, 2015.