Final Report

VedinkakSa (Summer 2019)

Teacher's side

On a successful app launch, after connecting the phone to UCCN Lab network, the app will launch the login screen for both students' and teachers' side. Teachers will have an id beginning with a substring ("IITG"), which will be checked at the login activity, and the user will be directed to the teachers' side, all other ids or roll nos. will be directed to the students' side.
Currently on the teachers' side, the Quiz part is fully functioning where a teacher can add questions to the database which will be further appeared in student's side for quiz activity, visualization part is also fully stable and functional with three different types of visualization, symbol based, image based and dynamic visualization, where as the classroom section is partially functional and few issues are to be fixed.

Teachers' credentials are to be provided by the admin, and hence they do not need not to register.

Student's side

The registration and login is same as of the teachers' side, the difference is that, to log in students need to scan a QR code, which consists of the position of the seats.
The program scans only valid QR codes, to ensure a certain level of validation of QR codes.
***While generating QR codes, please ensure that the JSON object is in the following format:
{"name":"SUBCODE","X":"<Xcoordinate>","Y":"<Y coordinate>"}
The QR scanner scans for the parameters "name", "X", and "Y".

The name must always have “SUBCODE” in it for the QR to be valid, “SUBCODE” can be changed to any other string by changing the string in method, onActivityResult(), in LoginActivity.

REGISTRATION AND LOGIN PROCESS

LOGIN

On launching the app, the LoginActivity pops up, which also has the link to register a new user, quoting ”new user? Register here”.
On Teachers side, Entering the data and clicking on the Login button is sufficient, but for students side, scanning a QR code is mandatory.
qr.initiateScan() starts the scanning activity, onActivityResult(), looks for the parameters in the QR code and validates.
Now, on clicking login, which calls attemptLogin(), which checks and validates all login credentials are input correctly or not and then calls JSONProcess2(), which collects the password and other data and compares the password fetched from the database with the user input password for the given roll no. if the password matches, login is successful otherwise rejected.

***Please note that on students’ end, on clicking login after JSONprocess2(), another method SendDataToServer() is called which sends the data like imageurl of the student to a table ‘image_tbl’ from where this image url is fetched for visualization purpose in students’ end. During Registration, the image url is stored in the table “images” along with roll, name and password. ***

The .php files used can be found in the server at:
172.16.117.43/AndroidImageUpload/
The .php files used in the Login process are
login.php fetches data like roll and pass from the table images for comparison.
qrupload.php saves the image url,roll etc in the table image_tbl.

For teachers side on logging in, truncating.php is triggered to truncate the table image_tbl, each time the teacher logs in.
REGISTRATION
On clicking fore new registrations in LoginActivity, the UploadPics activity is launched. There are text fields for roll, name and passwords, and a button to choose a picture of which url is talked about above.
The methods uploadMultipart() is responsible to upload the data like roll, name, pass and the picture, to the database.
The upload.php file is used to save the data in the table images.
Note all the tables are stored in the database db_images.
The next method in this activity is the FileChooser() which enables the user to choose the picture.

ATTENDANCE

Once the student is logging in successfully, automatically attendance will be updated in the database.
The database used to record this is android and within android database check the attendance table.

EXAMINATION

Examination activity has been added for quiz and exam purpose within classroom. On students' side student will see questions will four options students have to choose the correct option out it. At the end of the quiz total no of correct answers which is chosen by students will appear to the screen.

Database involved in this activity is "quiz". Within "quiz" database you will see a table by name of "question". All the questions are recorded there.

CLASS NOTES

A class notes activity has also been present there to record daily class notes for the student. Student has to type their notes after logging in. After saving the notes they can read the notes from their phone memory.

Class notes will be saved inside the "vedinkakSa" folder within it you will see classnotes subfolder, inside it all the saved notes will be present.

VISUALISATION

The classes, Downloader_one, JSONParser, PicassoClient, VisualisationConnector,VisualisaionScreen, VisualisationCustomAdapter etc. are responsible to carry out the visualization in the teachers' end.

For the dynamic visualization two classes has been added DynamicVisualization and DynamicVisualSecond.

The php files that support the visualization are truncating.php, student.php.

The states are fetched from the table images_tbl, and the states here are saved into the images_tbl by the file pyphp.php after prediction of the values of arousal and valence by the machine learning model classy.php located at the server in the folder
var/www/html/Train_and_test_files_arousal_classification/
**An IDE is installed to monitor the working of the python script ** and can be launched by executing the following command: ./pycharm.sh in the terminal.
Please note that the command above is to be executed inside the bin folder of the pycharm-community folder on the server.

PREDICTION OF AROUSAL, VALENCE, and EVALUATING EMOTIONAL STATES

On entering the classroom section from the students' side, there are two types of data collected, type features data and touch features data. Which are sent to the server by the help of uploadToServer.php in the server. The files(.csv files) are stored in the "new" folder on server. Now the python file classy.py is trained once and then is set to predict for all incoming files each 7 seconds for an infinite time until stopped. All the csv files are named as the roll no. of the student and hence storage of the values of arousal and valence and hence states are done against the roll nos. in the table State_Calc, and also to the table images_tbl, by the help of a php file pyphp.php.

Also note while registering for a new user, a separate table with roll as the table name is created and the states with timestamp are also stored in these tables with roll as tablename.

Finally, mentioning the task of all the .php files:

login.php -> fetches data to compare passwords during login from images table.
qrupload.php -> saves login data and url to images_tbl.
truncating.php -> truncates the table images_tbl, whenever the teacher logs in.
student.php fetches data from images_tbl for visualization in teachers' part.
upload.php -> Saves data in images table on a registration of a new user.
pyphp.php -> Saves predicted values of arousal, valence and emotional states to State_Calc table and images_tbl table.
*****Note that all the above mentioned .php files are inside the AndroidImageUpload folder.*****

The other .php files like uploadToServer.php are stored in var/www/html/
Two more classes need to be explained are Constants.java and Global.java.

These two classes hold all the urls for accessing the .php files and hence server. any changes in ip(s) or url(s) can be made there.
***Note that all the folders containing the .php files or storing the images or files(.csv, .txt, etc.) must be granted permissions, for which the following command can be used.
chmod 777 –R <folder/directory name>
Tables Used:
images -> Stores roll, name, pass, image url.
Uses: 1. Stores new data on registration
2. Data for login purpose is retrieved from here.
image_tbl-> Stores image url, and position and state for each student.

Uses-> Visualisation on teachers' end.
State_Calc -> Stores the different outcomes from the python models.

****

Partially developed part

On teachers' side as well as that of students' , the query mechanism is developed and is partially functional. As of now students can send the query the query but is only getting stored in database. For that the files queryupload.php is used, and the table queries is used.

****

***** **Before removing or making any changes in any part of the project or files, it is highly recommended to check for all the uses of that file. A small change or mismatch may cause error.*****