Speechless

Subhadip Nandi(16111023) Soumik Dasgupta(16111022)

April 30, 2017

Introduction:

Quite often in a big classroom setting students are hesitant to ask their doubts to the teacher. This hinders his understanding and ultimately he loses track of the subsequent topics as well. A periodic feedback is also useful for the teacher to be sure that his ideas are getting through to the students as intended. This app is our humble attempt to make the classroom experience more interactive and dynamic.

Although, there are several apps that try to remedy this but almost all of them require the students and teacher to be connected to the Internet, for the exchange of information throughout the duration. Since the standard routers can handle only so much, most practical systems fail to serve the intended purpose. This leads to students losing the network connection time and again. Our android app on the other hand circumvents this constraint by providing access to wifi to students in a round robin fashion, thus ensuring proper connectivity for all.

Model:

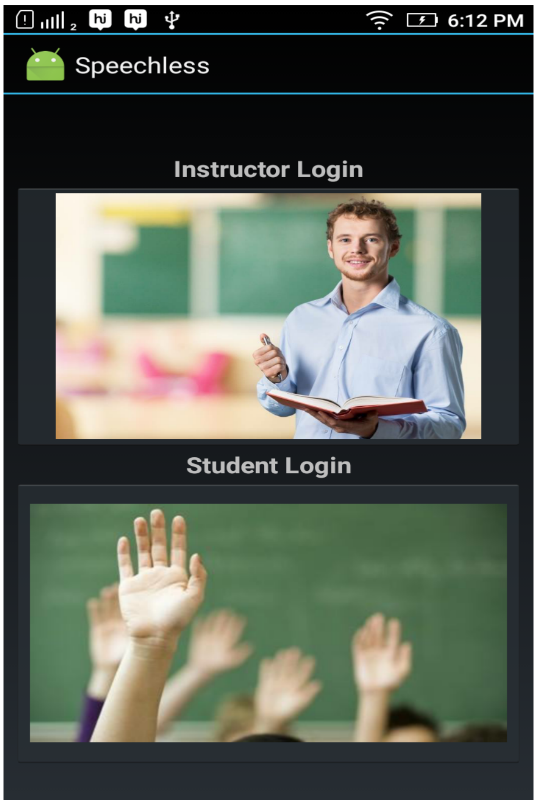
The class duration is divided into a number of sessions (defined by the instructor). At the start of each session, the students automatically get keywords for the topics to be covered in the next session (Download Phase). This gives the students a sense of context for the coming session. At all times during the session the students, have the keywords in front of them. Whenever a student feels out of pace with a topic he can cast a vote against the corresponding tag(Lecture Phase). At the end of each session, all the votes are aggregated and is displayed to the instructor(Collection Phase) and he can get back to the topics with which most of the class is struggling with and adapt his delivery for the subsequent sessions. At the start of the next session the students are presented with new set of keywords. The students can keep the tags and their votes as a later reference for themselves.

Interface:

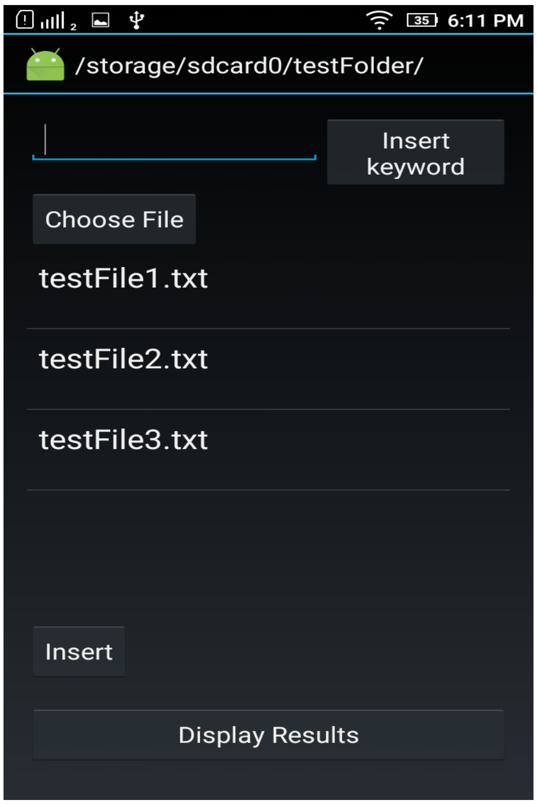
The figures on the next page are snapshots of our app in action. The Login Page is the page that shows up on starting our app.

The teacher page is where the instructor can upload his lecture slides from where keywords are extracted. Pressing the insert button sends these to the database where they are stored for students to access. Insert Keyword button allows the teacher to insert individual keywords not originally present in the documents. This gives the flexibility to the teacher to stray from the scheduled discussion if needed. After students have voted during the lecture phase, "Display Result" button at the end of the session displays the aggregate votes sorted with respect to maximum votes the teacher to get an understanding of the state of the class and adapt the level of teaching accordingly.

Figure 1

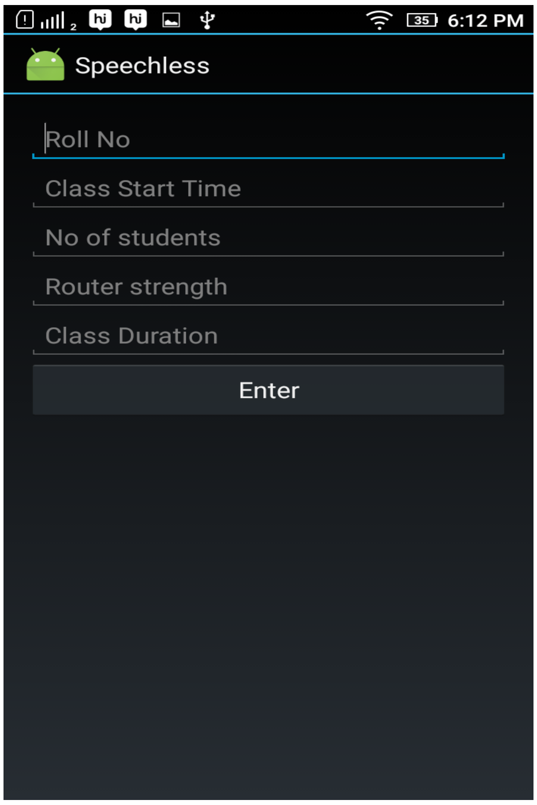


(a) Login page

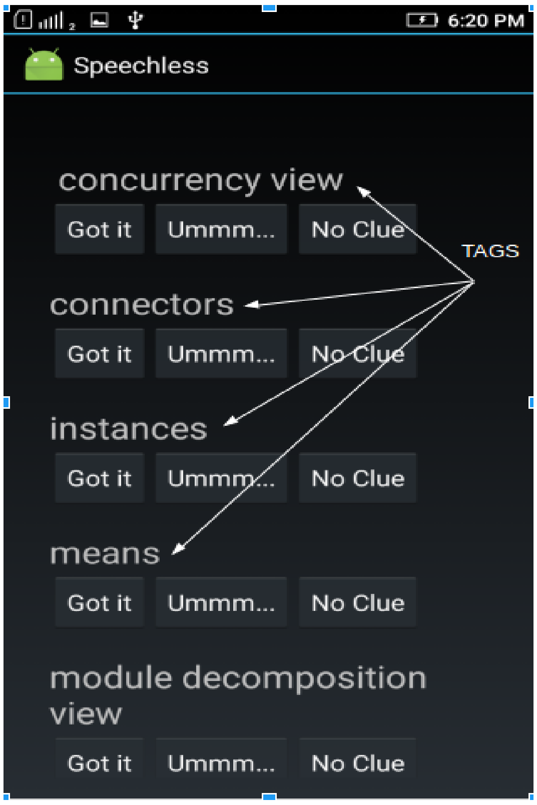


(b) Teacher page

Figure 2



(a) Student entry page



(b) Student page

Architecture:

The bottleneck phases are the ones where the students try to do server communication - download the tags and upload their responses(votes). To circumvent the limitations of the router, we use time multiplexing. We partition the entire class into groups based on their roll number and give each group slot at using the router. Each group contains a number of students which the router can handle simultaneously 4. Each slot is divided into three phases:

- Establish Phase: This phase establishes WiFi connection to the nearest router.

- Transfer Phase: This is the phase where actual data transfer occurs.

- Disconnect Phase: In this phase, the connection with the router is broken so that the next group of students can enter the establish phase.

Each phase is calibrated to synchronize with the class.

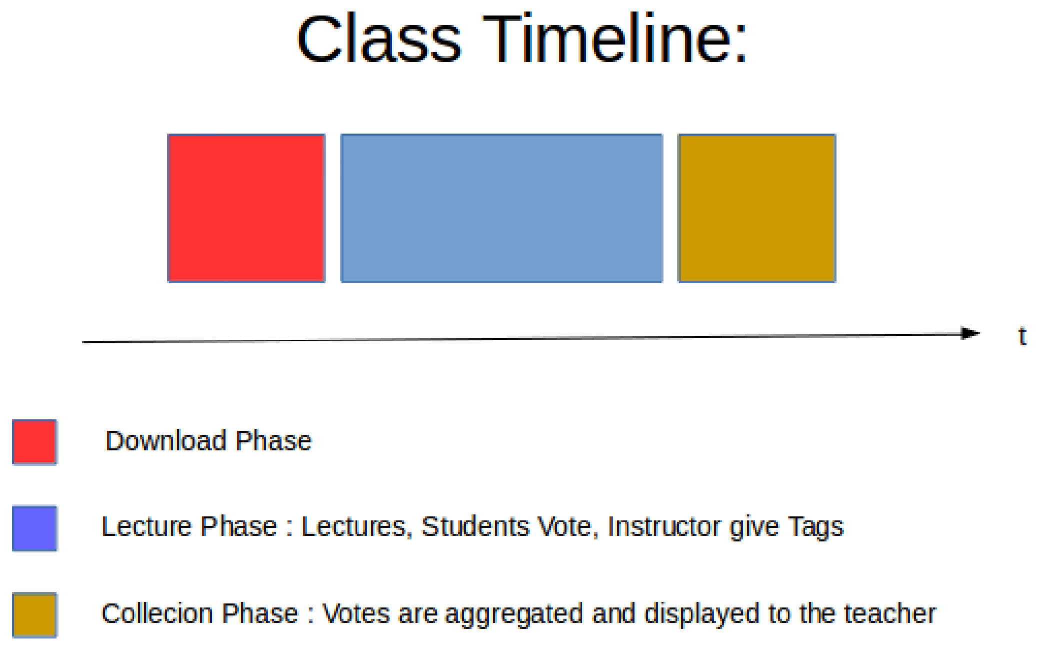

Class Timeline:
t
Download Phase
Lecture Phase : Lectures, Students Vote, Instructor give Tags
Collecion Phase : Votes are aggregated and displayed to the teacher

Figure 3: Class Time-line. Here the bottleneck phases are the Download and collection phases

Implementation:

The front-end for the app is written in java(android) with 4 activities, 2 for the student interface, 1 for the teacher and 1 for teacher/student module selection.
We have used IBM Bluemix Natural Language Understanding api in order to extract keywords from the documents uploaded by the teacher. The keywords then pop up on the students' side dynamically, who can then vote against these words. There is also provision for the teacher to enter keywords himself. This is important as the content being discussed in class may not always strictly adhere to the documents prepared by the teacher. Finally there is a button on the teacher's end to display the results obtained by averaging responses of all the students.
For the backend we have a php server running on localhost, containing mysql databases for storing keywords and student responses. The server responds to any get/put request made from the

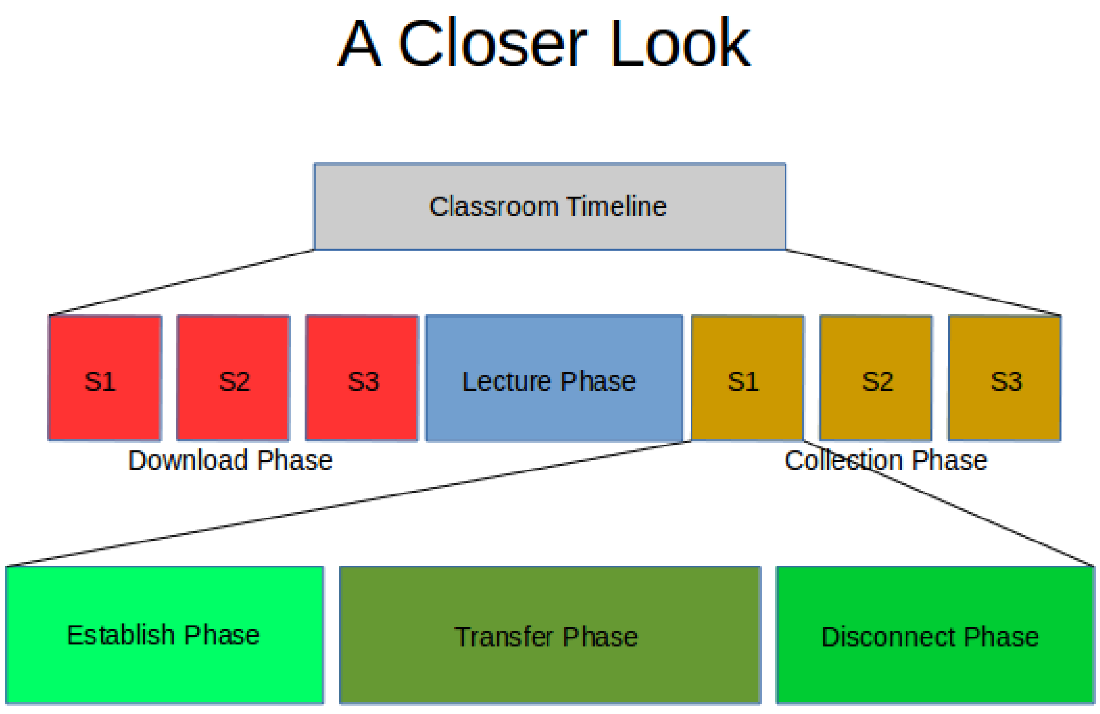


Figure 4: Architecture: The class is partitioned into student groups S_1, S_2, S_3,... and each group is given a turn in Round-Robin fashion

teacher or students' side.
In order to implement the time multiplexing as mentioned in the Architecture section, we have divided the students into different groups (S1,S2,S3,S4..), the strength of each group depending on the router capacity. We have called the inbuilt handler.postdelayed() method extensively throughout our code to delay each of the operations accurately so that none of the phases overlap.We have also given our app necessary permissions in the manifest file and created an instance of the WifiManager to turn the students' wifi on and off automatically during the establish phase and disconnect phase respectively, without the student having to explicitly do it everytime.

Future Scope:

In our work, we have assumed that students vote fairly, based on their understanding of the content. But, that may not always be the case. Some students may intentionally vote some topics highly so that the teacher has to repeat the same topics again and again. In order to prevent this we have thought of adding a quiz at the end of each session. Students performing well in the quiz will be given higher weights. If high scoring students vote highly against some content, then there is be a clear need to revisit those areas of the class. Similarly if the votes cast by students performing poorly reflect that they have understood some topic, then there is no need at all to revisit those areas.

Acknowledgement:

We would like to thank Professor TV Prabhakar, Saurabh Srivastav, Chetan Gupta and Sumit Kalra for their guidance throughout the course of our project.