50.001 Team 3 - 8 1D Report

CHEN YAN 1003620 | WANG TIAN 1003769 | HE YUHANG 1003775 | WANG ZILING1003783 | LEO DING HAO 1003510 | WONG KAI KANG 1003625

Problem Framing

Most classrooms do not have a lecture capture system. Even if it is installed, only very few classes and instructors use it. At the same time, we found that lecture videos are not presented in a way that best assists students' learning process. To find a certain point covered in class, students might need to sieve through the video and type the point word by word into their notes. This is time-consuming and the students may therefore miss important content. Hence, in order to maximise students' learning and help them to improve learning effectiveness, we made this project.

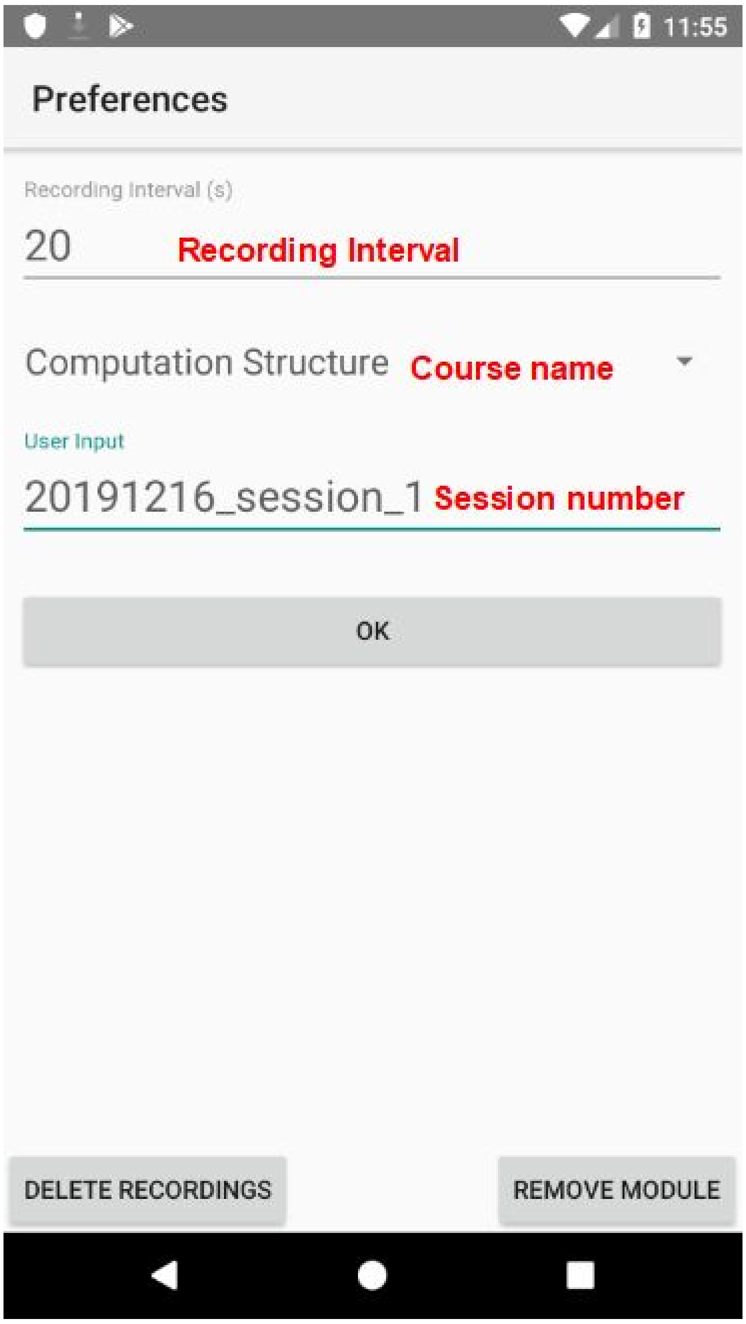
Solution

Firstly, as the pace of the classes is fast, it is hard to type down all the key points into their notes. Our app will provide a way to help students take notes easier. At the same time, we added a support search function in order to decrease the time that they waste to find certain topics covered in class.

Our main idea is to transcribe every word the instructor said and written on the whiteboard into texts using state-of-the-art technologies. By transcribing handwriting and speech into text, we can support search function and make it possible for students to directly copy texts into their notes, and therefore provide students with an interactive, convenient and engaging learning experience.

Recorder App

This app is dedicated to record the lecture and send the recordings to firebase storage and update data in firebase database. Users have to set two values before recording which are course name and session number in that day. One additional setting is recording interval which determines recording video segments length and segments are sent to firebase in background piece by piece.


Preferences

Recording Interval (s)

20 Recording Interval

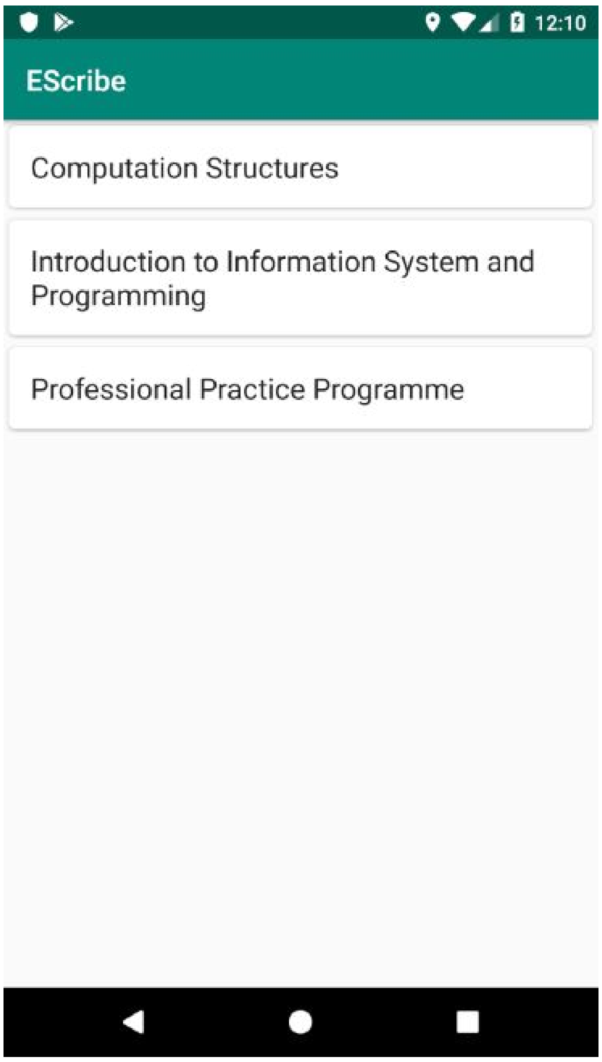
Computation Structure Course name

User Input

20191216_session_1 Session number

Viewer App

This app is students to view the recordings by course name and session number and post their questions, doubts, ideas and thoughts. Students can have discussions under the video so they can have a better understanding of the course.
We have set three levels in this app where the first level is course selection. The course names in the firebase are displayed here.

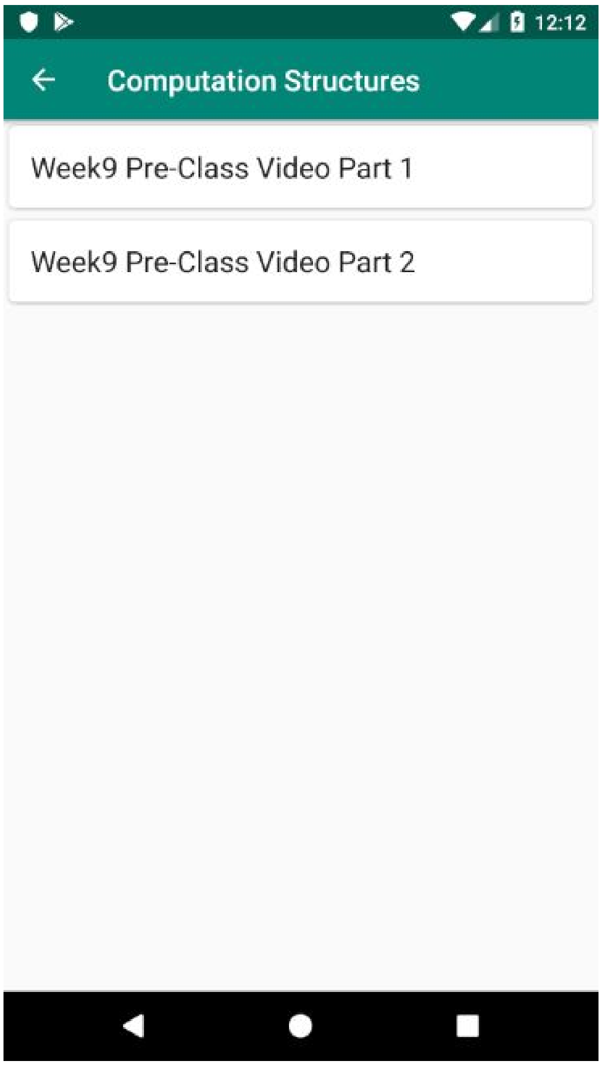

12:10

EScribe

Computation Structures

Introduction to Information System and Programming

Professional Practice Programme


12:12

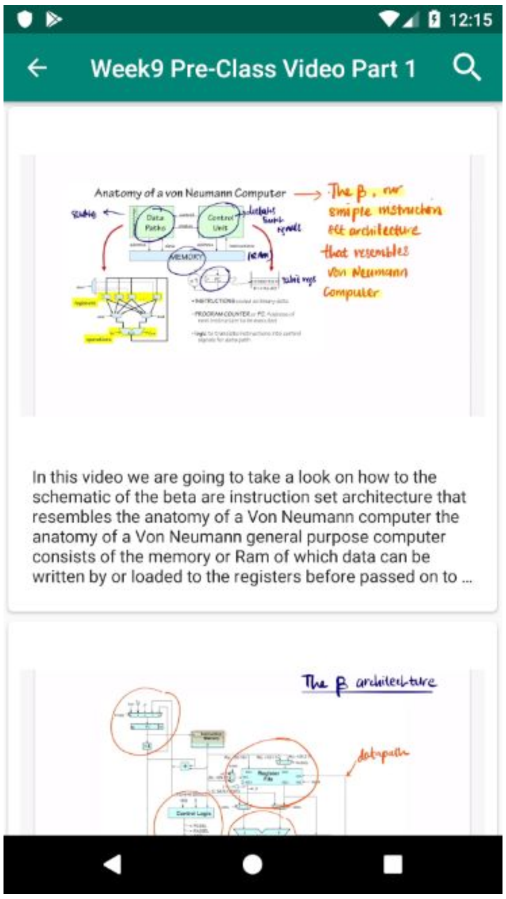
Computation Structures

Week9 Pre-Class Video Part 1

Week9 Pre-Class Video Part 2

The next level is session number selection, student can choose which course he wants to watch.

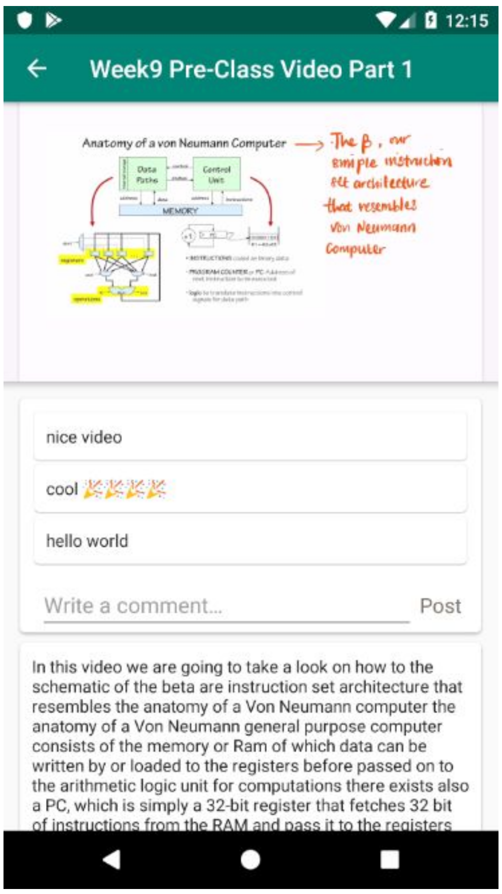
Lastly students can choose video parts splitted by lecturer's different slides. Each part is coming with slides thumbnail and speech recognition text. Then students can click certains parts to watch the video, post ideas and view speech recognition text and slides recognition text. There is also a search function where students can type in what they want to find and the app will do a search in recognition texts and return the corresponding videos.


12:15

Week9 Pre-Class Video Part 1

Anatomy of a von Neumann Computer
The β, our simple instruction set architecture that resembles von Neumann Computer
The β architecture
datapath

In this video we are going to take a look on how to the schematic of the beta are instruction set architecture that resembles the anatomy of a Von Neumann computer the anatomy of a Von Neumann general purpose computer consists of the memory or Ram of which data can be written by or loaded to the registers before passed on to ...


12:15

Week9 Pre-Class Video Part 1

Anatomy of a von Neumann Computer
The β, our simple instruction set architecture that resembles von Neumann Computer

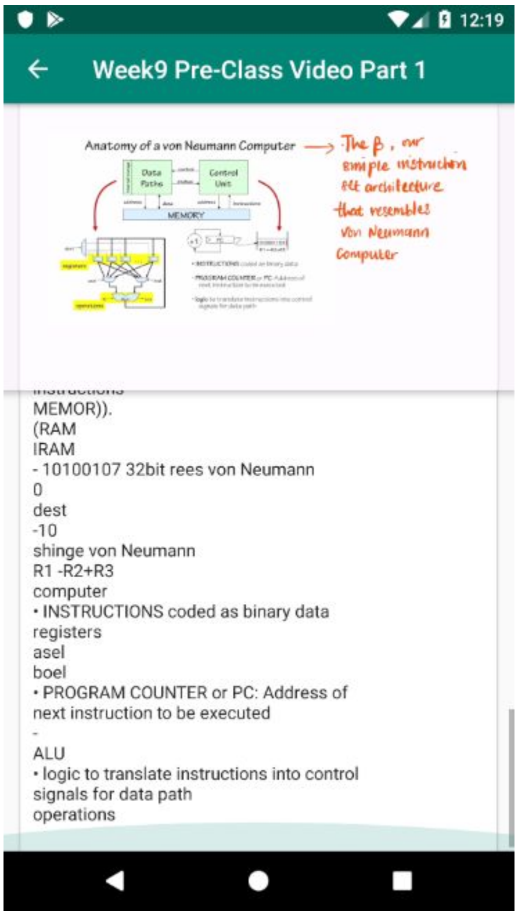
nice video

cool 💥💥💥💥

hello world

Write a comment... Post

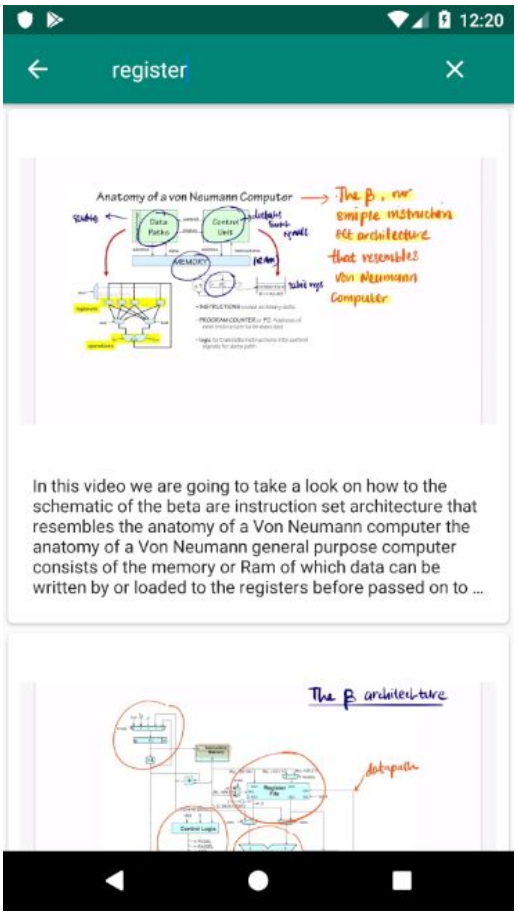
In this video we are going to take a look on how to the schematic of the beta are instruction set architecture that resembles the anatomy of a Von Neumann computer the anatomy of a Von Neumann general purpose computer consists of the memory or Ram of which data can be written by or loaded to the registers before passed on to the arithmetic logic unit for computations there exists also a PC, which is simply a 32-bit register that fetches 32 bit of instructions from the RAM and pass it to the registers


12:19

Week9 Pre-Class Video Part 1

Anatomy of a von Neumann Computer
The β, our simple instruction set architecture that resembles von Neumann Computer

MEMOR)).
(RAM
IRAM
- 10100107 32bit rees von Neumann
0
dest
-10
shinge von Neumann
R1 -R2+R3
computer
• INSTRUCTIONS coded as binary data
registers
asel
boel
• PROGRAM COUNTER or PC: Address of
next instruction to be executed
-
ALU
• logic to translate instructions into control
signals for data path
operations


12:20

register

Anatomy of a von Neumann Computer
The β, our simple instruction set architecture that resembles von Neumann Computer
The β architecture
datapath

In this video we are going to take a look on how to the schematic of the beta are instruction set architecture that resembles the anatomy of a Von Neumann computer the anatomy of a Von Neumann general purpose computer consists of the memory or Ram of which data can be written by or loaded to the registers before passed on to ...

System Architecture

Overview

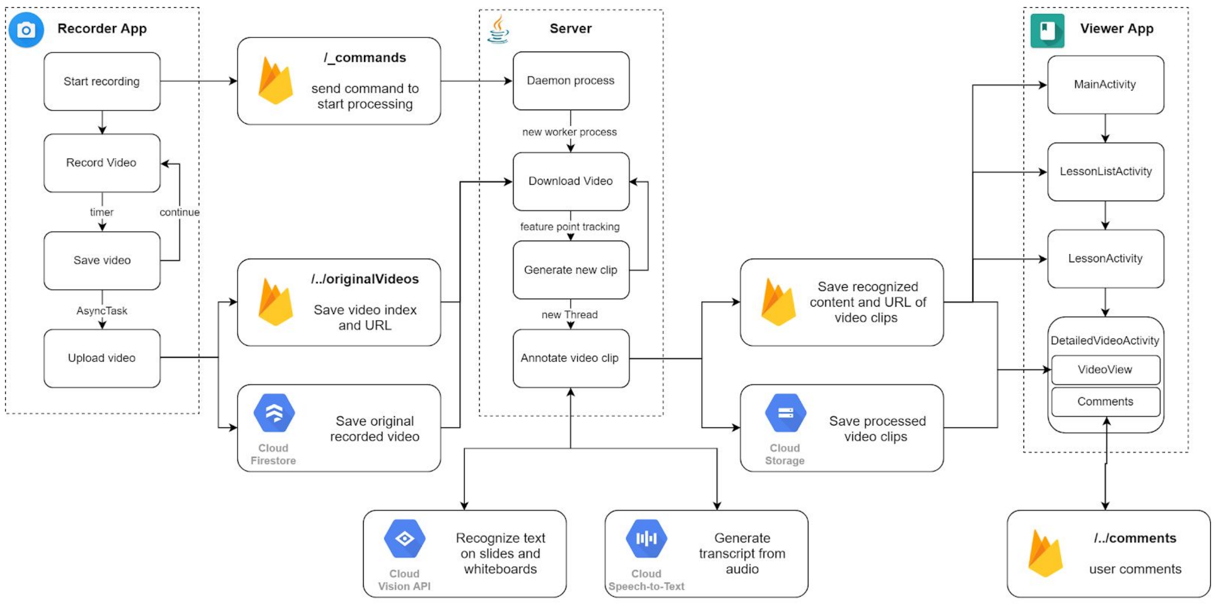


Figure 1: Data flow in the system

The system consists of three components. There is a recorder app which records lecture videos and uploads to the cloud. Then, the original videos are downloaded to our server program where indexing and content recognition happens. After processing, there is a viewer app for students to easily review the class and have discussion.

The goal of our system is to be reliable and efficient. Therefore, Firebase Realtime Database and Google Cloud Platform are chosen as the intermediaries between our components. With Firebase's async API and widely used async tasks in our system, everything can happen in real-time efficiently and reliably.

The dataflow of our system is shown in *figure 1* above. It begins with the recorder app. When the user clicks the start recording button, a command is appended to a Firebase commands queue to notify the server to start processing. A database listener is set in the server's daemon process which will start a new worker process to continuously load video from cloud and process it. Then the recorder app will begin recording. After a certain period of time (which is configurable by user), a new video piece will be saved and an async task is launched to upload that piece to Firebase Firestore. And the recorder app will continue recording. On the server-side, a listener is called every time a new video is uploaded in the worker process. Then the video is downloaded and combined with previous videos (soft linked, no encoding required). Meanwhile, a feature tracking algorithm using OpenCV runs and will generate a new video clip when the course subtopic changes. The video clip will be encoded and uploaded to Google Cloud Storage. Audio channel and keyframe are extracted and annotated using Google Speech-to-Text API and Google Cloud Vision API. Video URL and recognized content are then uploaded to Firebase. Students can use our viewer app at any time, even while the class is happening. The app reads most updated data from Firebase in realtime using the firebase listener. There is also comments function for students to have a discussion, which is saved in Firebase as a String list.

Recorder App

The Recorder App uses the android.hardware.camera2 package and the MediaRecorder class for video capturing. The base structure of the code is taken from the Android Camera2Basic Sample repository and modified accordingly. At first, a CameraManager instance is initialized to retrieve the properties of the camera in the form of a CameraCharacteristics object. From the CameraCharacteristics object, we are then able to get the resolution sizes for the camera preview and video recording which will allow us to set the dimensions for the TextureView in the activity. At the same time, a MediaRecorder object is also initialized to be used later for the audio and video recording. From the CameraManager instance, the openCamera method will be called to establish a connection to the camera, if the camera successfully opens, the camera preview will be started as specified in the StateCallback provided. When the recording button is pressed, the appropriate sources and encoding will be set for the MediaRecorder object and the prepare method will be called. Through the FirebaseActivities instance, a processing command will be sent to the command queue in the realtime database. A CaptureRequestBuilder will then be created through a factory method in the CameraDevice instance obtained from the StateCallback previously. Surfaces for both the TextureView (Camera Preview) and the MediaRecorder will be added as output targets for the CaptureRequest through the builder. A CameraCaptureSession will be created through the CameraDevice instance and the CaptureRequest instance will be passed into it as a parameter. The MediaRecorder will then start to record whatever stream is passed through the target surface as declared in the CaptureRequest. At the same time, a Handler object will be initialized with an anonymous class implementing the Runnable interface to invoke the record button click after the set time interval. This is to simulate a recording loop until there is further input from the user. When a video has been recorded and saved to local storage, it will be uploaded through the FirebaseActivities instance. Upon a successful upload, the corresponding download link will be updated in the database.

Video Processing Server

In the video processing server, video pieces from recorder app are downloaded and combined by FirebaseVideoReader. The screen or whiteboard area is then cropped using an algorithm that locates the largest rectangle in the frame. Then the feature point tracking algorithm runs for every two seconds and generate a new clip when features changes.

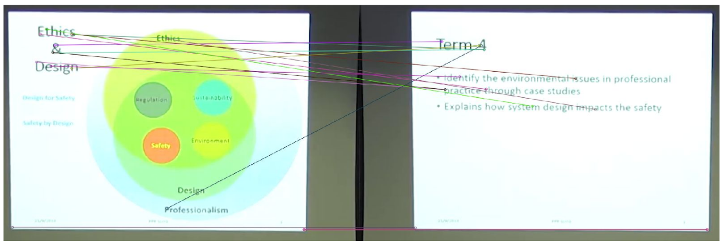

Ethics
&
Design
Ethics
Regulation
Sustainability
Safety
Environment
Design
Professionalism
Term 4
• Identify the environmental issues in professional practice through case studies
• Explains how system design impacts the safety

Figure 2: example of two discontinuous frames

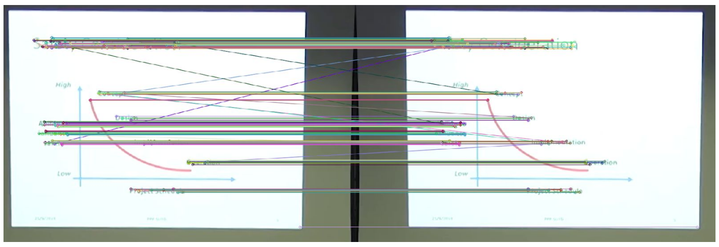


Figure 2: example of two continuous frames

After the video clip is generated and encoded, it is further annotated using Google Vision API and Speech-to-Text API. And the results are uploaded to Firebase.

Viewer App

Three layers indexing (course -> lesson -> slide) is used to allow students to quickly locate content that they want to view. Each layer is implemented using an Activity with RecyclerView. And intents are used to launch the next layer activity. In the detailed video view, a VideoView is used to play recorded video.

Effective Programming

Adapter Design Pattern

The Adapter Design Pattern is implemented through use the spinners in the Recorder App. An ArrayAdapter class is used to facilitate the interaction between the Spinner widget class and the object containing data which is an ArrayList of strings.

Adapter Design Pattern is also widely used in viewer app to map data in Firebase Realtime Database to RecyclerView. Data is firstly added to a List by Firebase list. Then in the RecyclerView adapter, item views are inflated using list data.

Observer Design Pattern

Observer Design Pattern is used in the viewer app to update list items. When the list is edited by Firebase event listener, notifyItemInserted() method is called to notify the RecyclerView, which is the observer, to update the UI.

Inheritance, Polymorphism and Abstract Class

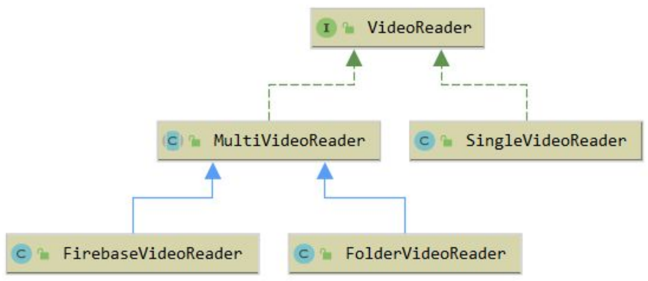


Figure 4: UML diagram of videoreader package

When developing and debugging the video processing program, one common problem is that the video capture can come from multiple sources. During debugging, the VideoCapture class provided by OpenCV may be enough. In production, however, the program is supposed to read video pieces from Firebase and provide a continuous frame reader to downstream processes. However, we do not want the downstream program to care about what exactly the video source is. Therefore, a common VideoReader interface is defined. Then a SingleVideoReader implementation which also inherits OpenCV's VideoCapture is used for debugging. An abstract MultiVideoReader implementation is then defined to combine multiple SingleVideoReaders (provided by abstract method) to a single VideoReader. Then two subclasses FirebaseVideoReader and FolderVideoReader are written, one reads video from the Firebase, and another one from a local folder.

Single Responsibility Principle

For the recorder app, the process of uploading of the video file to firebase storage is encapsulated in a class of its own rather than in the main camera activity class. The Camera2VideoFragment class retains its initial responsibility which is to record the stream from the Camera while the task of handling Firebase references and UploadTasks is delegated to the FirebaseActivities class.

Team Allocation

Recorder App	Leo Ding Hao, Wong Kai Kang

Server	He Yuhang
Firebase	Hua Guoqiang
Viewer App	He Yuhang, Wang Zilin, Chen Yan, Wang Tian, Hua Guoqiang
Poster	Wong Kai Kang
Exhibition	All members

Conclusion

In conclusion, we successfully implemented all the functions that we want. All the parts work well. When the video plays, it transcribes every word that the instructor said accurately and written on the whiteboard into texts. It is quite convenient for students to directly copy words that instructor said or written on the whiteboard into their notes. In addition, the recorder of our app is also working well, it implements storing, retrieving and displaying information effectively so that the support search function can accurately work by searching keywords.

Through this 1D project , we have gained greater appreciation of the difficulty and learnt a lot about development in Java and Android. Thanks for everyone's efforts and contribute to our projects. Especially , we are very appreciative of our leader He yuhang . He is extremely capable and brilliant leader. He is extremely hardworking,intelligent and contribute a lot to our projects and give us help in technical problem.

Reference

https://developer.android.com