Android Video Player Application

#Write a video player application with ‘Play’, ‘Forward’, ‘Rewind’ functionalities. Please write pseudocode for this program and explain the design pattern you will use to develop these three functionalities.

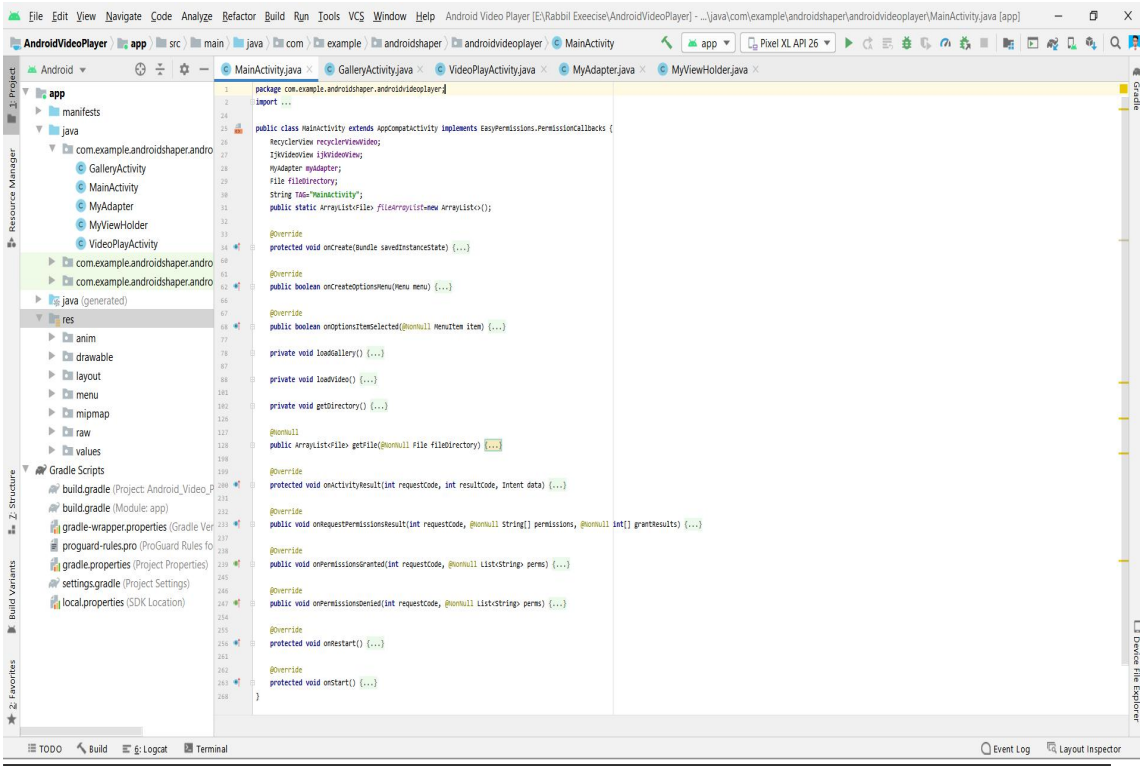
#Answer to the question of video player 3 functionalities pseudocode and design pattern.

Here i show your phone all videos in a recyclerView likes a video library if you click any video then start this video if you want go next video then click next button then path ArrayList position will be change to the next path and play next video when you want go previous video then video will be back. Here Play, forward, rewind, pause etc all features give this application.

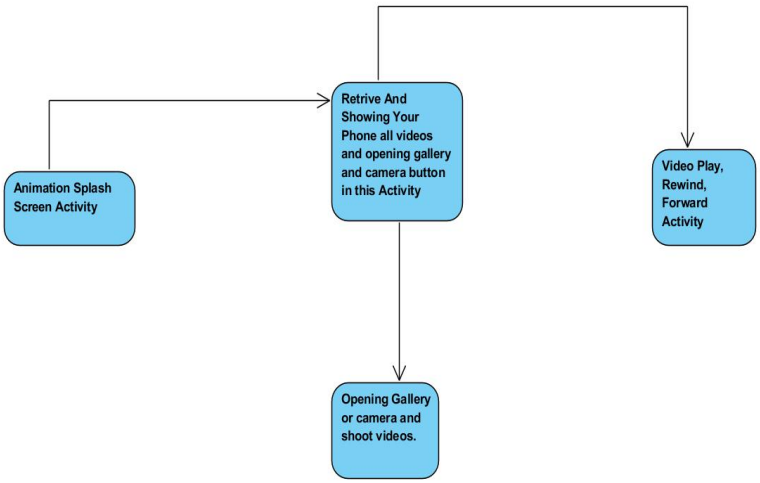
In android using VideoView component and MediaController class we can easily implement the video player in android application to player the videos with multiple playback options, such as play, pause, forward, rewind etc.

Generally, the MediaController class in android will provide playback options for video player, such as play, pause, forward, rewind etc. We also work this MkVideoPlayer class but there i only work this task in MediaController class and also give example pick video in gallery by MediaPicker. When we pick video path then we complete requirement functionalities by this MKVideoPlayer class and we also shoot video this MediaPicker class object.

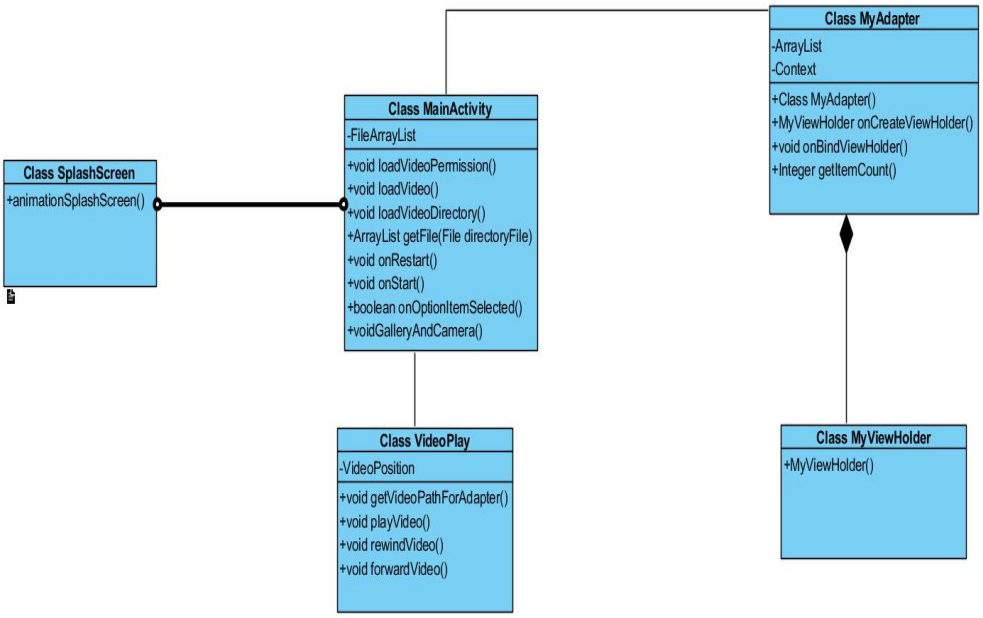
Project Screen Shot



Application Activity Diagram



Application Class Diagram



MainActivity.java

#Here getFile() method which make a ArrayList to video file path .

```
@NonNull
public ArrayList<File> getFile(@NonNull File fileDirectory) {
       if (fileDirectory.isDirectory())
       {
           final File[] files;
           files=fileDirectory.listFiles();

           if(files!=null&&files.length>0)
           {
               for (int i=0; i<files.length; i++)
               {
                  if (files[i].isDirectory())
                   {
                       getFile(files[i]);
                   }
                    else if (files[i].isFile())
                   {
                       if (files[i].getName().endsWith(".mp4"))
                       {
                           fileArrayList.add(files[i]);
                           for (int j=0;j<fileArrayList.size();j++)
                           {

                     if (fileArrayList.get(j).getName().equals(files[i].getName()))
                               {

                                   Log.e(TAG, "getFile: "+"Same name file");
                               }
                               else
                               {
                                   fileArrayList.remove(files[i]);

                                   fileArrayList.add(files[i]);
                               } } }} }}}
       else
       {Toast.makeText(getApplicationContext(),"Video not available",Toast.LENGTH_SHORT).show()}
       return fileArrayList; }
```

#Here getDirectory() method Pass ArrayList in MyAdapter for showing my all videos.

```
private void getDirectory() {

       String  rootPath=  String.valueOf (Environment.getExternalStoragePublicDirectory
(Environment.DIRECTORY_DCIM));
       fileDirectory=new File("/mnt/");
       fileArrayList= getFile(fileDirectory);
       Log.e(TAG, "ArrayListSize: "+String.valueOf(fileArrayList.size()));
       if (fileArrayList.size()>0)
       {
           myAdapter=new MyAdapter(getApplicationContext(),fileArrayList);
           recyclerViewVideo.setAdapter(myAdapter);
       }
       else
       {
           Toast.makeText(getApplicationContext(),"Arraylist is
Faka",Toast.LENGTH_SHORT).show();
}
}
```

VideoPlayActivity.java

Here VideoPlay() method where play,rewind,forward functionalities.

```
     private void VideoPlay() {
    MediaController mediaController=new MediaController(this);
    mediaController.setAnchorView(videoView);
    videoView.setMediaController(mediaController);

videoView.setVideoPath(String.valueOf(MainActivity.fileArrayList.get(position)))
;
    mediaController.setPrevNextListeners(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            int check=MainActivity.fileArrayList.size();
            if((position+1)!=check)
            {

videoView.setVideoPath(String.valueOf(MainActivity.fileArrayList.get(position=p
osition+1)));
                videoView.start();}
            else
            {

videoView.setVideoPath(String.valueOf(MainActivity.fileArrayList.get(position=c
heck-(position+1))));
                        videoView.start();
            }}
    }, new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (position!=0)
          {

videoView.setVideoPath(String.valueOf(MainActivity.fileArrayList.get(position=position-1)));
    videoView.start()}}});
    videoView.requestFocus();
    videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
        @Override
        public void onPrepared(MediaPlayer mediaPlayer) {
            videoView.start();}
    });

    videoView.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {
        @Override
        public void onCompletion(MediaPlayer mediaPlayer) {

videoView.setVideoPath(String.valueOf(MainActivity.fileArrayList.get(position=position+1)));
            videoView.start();}});
    videoView.canSeekForward();}
```