郑雅文：

1 Linux端口的MySQL配置

①轻量级服务器配置MySQL Linux端口访问该服务 服务器要过期了

②Linux端安装MySQL -->需要编译驱动 已采取该方式（后续改成第一种 ）

③自己继承了的sqllite 不好用

最终采取②

对于②，由于qt本身不带数据库相关的文件，所以需要自己编译链接。感觉还是比较麻烦。先下载源码，再对mysql的驱动进行编译，编译生成的debug和dll放在编译器下这样子可以实现数据库和qt的连接

2 实现qt对数据库的curd（c加加与mysql的交互）

测试mysql与qml的互相作用，已测试成功。

注册到main.cpp中，在mysq.h和mysql.cpp中写出curd的逻辑，并且显示出来。

3 将功能集合在一起

功能：如果点击发送弹幕的按钮，就实现了一条弹幕的发送

在数据库中建立一个库，保存对应的视屏弹幕信息。（用数据库也是为了后续增添登陆功能）

每一个视屏文件对应的弹幕信息保存在一张表里面。

对于这个表的字段

name 对应的谁操作的 varchar（10）

context 长文本

timestamp 时间戳，在视屏对应的位置进行显示。

插入的时候，会自动记录时间戳和内容，获取时间戳的顺序和弹幕的实现逻辑一样。

要进行事务管理，防止脏读和幻读以及不可重复读。

要建立索引？对时间戳建索引，提高效率。（后续删除了，因为时间戳生成是实时的,建立索引消耗蛮大的。）

要进行用户管理 （登陆？后期引入登录界面再拓展，先实现一个粗略的后端逻辑）

逻辑：

1qt和c加加数据互通

2点击文本框输入，点击发送或者按下空格的时候进行事件的检测，将文本框中的数据传送到c加加中，构造insert的记录，插入到数据库中。

3显示一直在检测是否有当前的内容，对于这种直接输入发送的，就直接从数据库中取出来并且显示到屏幕上 。

显示的时候构造qstiringlist，利用index输出 方便下一次继续插入。

但是后续由于无法将数据库的结果qstringlist返回构建字符串，并且构建了字符串也无法实时传回数据库。这样的思路就被抛弃了。改用xml文件。

实时弹幕

逻辑：这些数据都是要在后台的数据库中存储起来的

之后如果想根据视屏播放的视屏祯所处的位置来获取一段时间范围内的数据并且显示 （上限一次100条吧）

状态的检测是实时的

后来这样子做的话感觉不太好，用xml文件比较好实现。组员写了形成xml的方法。我进行读取文件并且构建字符串显示。但是时间问题以及组员形成xml文件格式有点问题，这个暂时没有实现。就只有显示弹幕的效果。数据来源并没有处理好。

李昕航：

1. PlayListModel和RecentFilesModel的思路（前期准备，不包括缩略图和字幕部分）：

PlayListModel 与RecentFilesModel的结构大体相似，我主要介绍PlayLisyView 和 最近打开文件是如何被记录下来的。

首先是 设定Item的内容，如果要实现打开一个文件，能索引同目录其他文件，就需要有文件的位置（Path），所在目录(Folder),以及ListView所要展现的文件标题（Title），视频时长（Duration）。

前面三个，应该使用QFileInfo去获取文件的绝对位置，和目录，标题。

视频时长我本来之前用的MediaInfo的库，但是很繁琐，需要将QString与wstring来回切换以及也需要外部依赖，虽然实现了但是效果不是很满意，后来我直接使用Qt自带的MediaPlayer类去获取时间，但是这个也有一些问题（如：介绍视频所说的MediaPlayer无法兼容有些视频的格式，无法获取时间）。

然后就是继承QAbstractListModel的PlayListModel如何去完成需求。

Model的需求有几点，一个是能打开文件可以选定对应的模式单个打开（appendItem）以及索引兄弟（getSiblingItem），然后是模式的切换bool，（最后由于qml设置界面没有完成，我就设置了默认索引兄弟，没有提供切换的选择）

如何索引兄弟，首先确认打开的文件存在，然后使用迭代器QDirIterator去历遍文件，

如果文件为视频文件（通过mimeType进行确认为”video/”）,则添加到StringList。历遍完成后，进行排序，添加到Item中。函数名 getSiblingItem()

添加单个文件类似，打开文件可以根据bool进行两者的切换，同时添加单个文件的方法可以在索引兄弟之后进行使用，继续添加在ListItem中(图形化界面中未展现)

RencentFile与其相似，多的一点是，利用QSetting进行存储最近打开过的文件，已有的不会被重复添加，打开之前的会将其提至最前。函数名updateRecent()

1. Srt字幕思路

字幕的思路本来是使用qml中MediaPlayer自带的播放机器，但是效果很差，卡顿，识别不完全。

后来我搜索字幕的解析，他可以将字幕分段提取出来，存储到类中，可以随着播放视频的时间，有对应时间段的字幕，在qml上就可以获取字幕text,并且通过Text展现出来，通过对qml text的设置可以进行移动（实现了被动移动）对应文件srtparser.h ,qsubtitleprovier.h/.cpp

但是只支持srt字幕。

然后ListMdoel的Item 添加字幕List,他可以搜索同目录同名字幕文件，也可以打开外部字幕。

1. Thumbnail 缩略图

利用ffmpeg 截取 视频中段的某一帧,将这一帧的Image存储到ModelItem中，然后使用QQQuickImageProvider，将C++的QImage传送到Qml中，同时Provider需要设置对应于Title的QImage。为了能够实现RecentFileItem与PlayListItem实现无差别，将两个进行抽象，再泛化，Provider对抽象类进行操作，实现两个ModelItem可以同时使用同一个Provider类。

对应文件名modelimageprovider.h listitem.h

1. 主界面和播放界面的跳转交互

因为主界面和播放界面的切换使用的是stackView,切换的Item之间没有办法直接进行通信，因此在stackView设置property进行信息交换，点击某一视频缩略图，将其path信息和index信息存放至stackView的property中，视频播放界面当其Compent.onComplete时进行设置实现交互

1. 优化qml中的界面布局，实现音量，字幕控件

优化视频播放界面的qml布局，使得视频控制条区域更加自由，hovered时显现的控件可以在视频播放区域进行操控

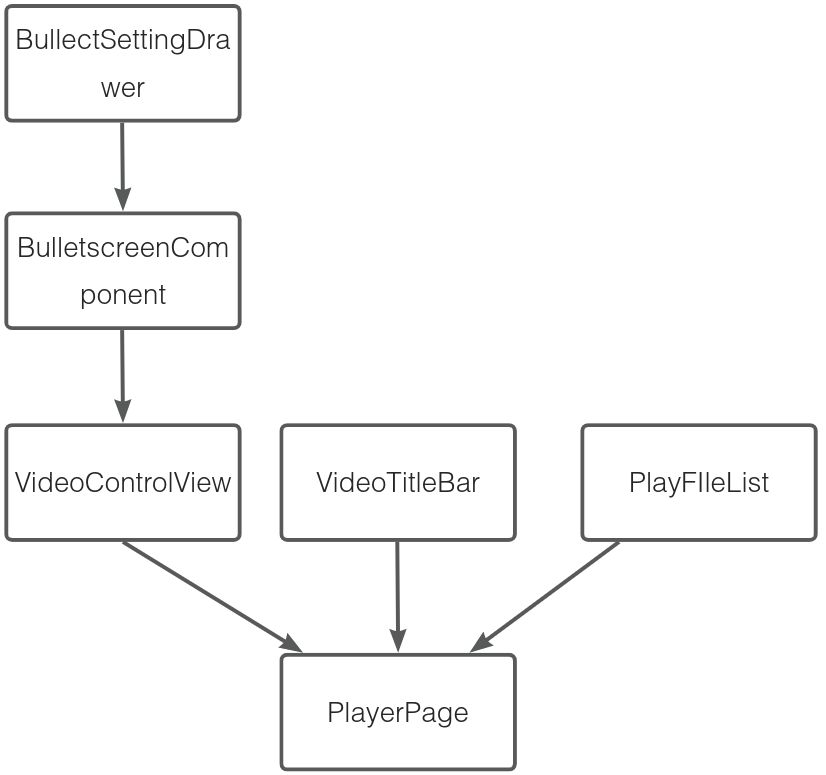
范榆康：

QML界面实现

一、层级关系

* 组件放在components文件夹下
* 逻辑实现放在logic文件夹下
* 模块放在model文件夹下
* 基本页面放在pages文件夹下

二、qml视频播放界面基本实现逻辑



三、播放界面的实现

播放界面分为三个层级，如图1

* 上方为VideoTitleBar视频标题栏，用于展示视频的标题。其中功能包含：返回上一级按钮。获取当前正在播放视频的视频名称，并显示。收藏按钮，文件列表按钮。整个视频标题栏有鼠标拖动区域，使用PointHandler实现鼠标拖动。右上角有鼠标缩放区域，使用PointHandler实现窗口的缩放。
* 下方为VideoControlView视频控制栏，左下方三个按钮分别是快退、播放/暂停、快进。弹幕条。字幕开关。倍速按钮。音量按钮、设置按钮全屏/缩小按钮。其中有组件计时器，当鼠标不动300ms后隐藏。
* VideoControlView视频控制栏中最主要为进度条。左右时间为Text，通过获取视频的时间进行转换，实现播放时间和时长的效果。进度条Slider重写了background，更改背景样式。
* 弹幕条BullectscreenComponent由弹幕开关，弹幕设置，弹幕输入组成。输入的弹幕点击发送后保存为本地与视频同级同名的xml文件中。弹幕设置按钮鼠标悬停即可出现弹幕设置框，可设置弹幕的大小和颜色。
* 中间为视频播放界面，TapHandler实现单击播放/暂停，双击全屏/缩小。onPositionChanged来判断鼠标是否在移动。当鼠标在移动时，所有计时器（标题栏显示、控制栏显示）停止，光标正常。当鼠标悬停后计时器启动，时间到隐藏对应的组件，光标空白。