SOA 大作业之

技术文档

班级： 计 34

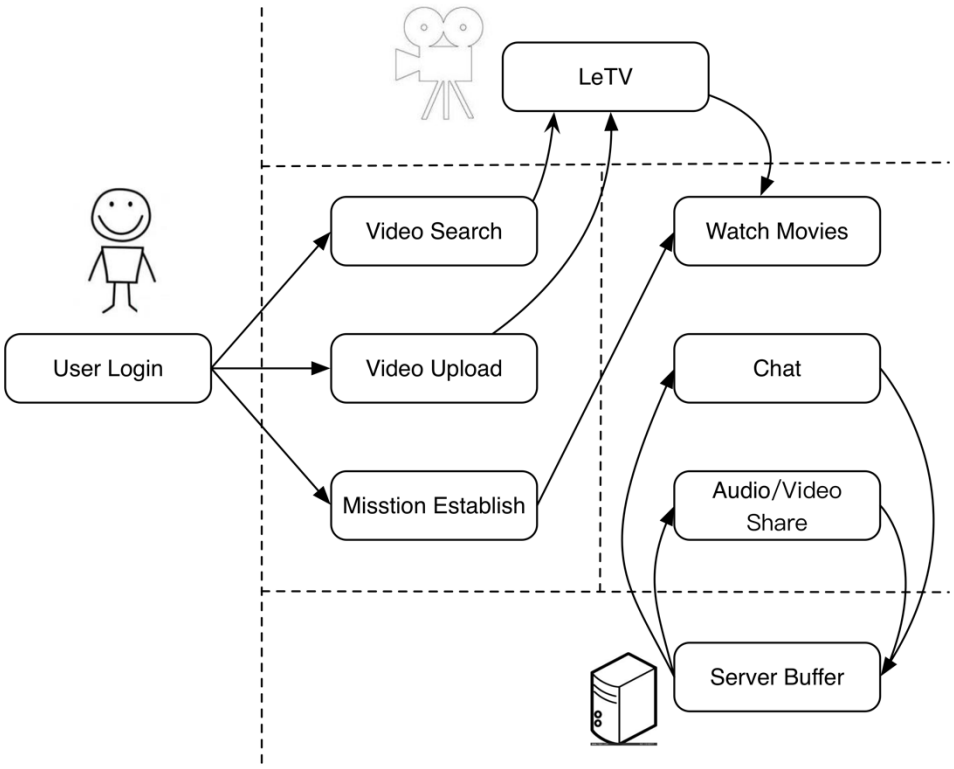
成员： 何晓楠 2013011361

黄予 2013011363

杨晓成 2013011383

一、总体架构

1. 数据流分析



2. 网站搭建

a) 网站使用 node.js 的 express 框架搭建

b) 数据库采用 mongodb

c) 在线聊天系统使用 socket.io 实现包

d) 视频的搜索和播放功能采用乐视 api

e) 前端使用 bootstrap 模版

3. 模块设计

a) 用户管理模块：实现用户注册、登陆、注销，实现好友的添加与删除。其中可能还包括邮箱验证。

b) 任务管理模块：实现建立观影任务、删除观影任务，更新观影任务，其中删除

与更新要顾及到所有观影成员的任务队列。

c) 搜索模块：实现视频的关键字检索，输入视频名称，返回符合的视频列表，供用户进行选择。

d) 视频管理模块：实现视频的上传，能够编辑已上传视频的名称和标签

e) 播放管理模块：实现视频的同步播放。

f) 交流模块：实现文字、弹幕、语音、视频（待定）的实时交流。

二、功能架构

1. 异地同步观影

用户在我们的网站上选择一个电影/电视剧，设定播放的时间，选择其他可以一起观看的用户，网站则新建一个播放任务，用户通过点击超链接进入观影页面。我们将在用户设定好的时间播放电影，一旦播放，不会暂停，后进入观看的用户只能从已经流逝过的时间开始观看，即用户的观影在时间上是同步的，相当于一个小众的直播。

2. 用户管理

a) 提供登陆，注册，和登出功能，部分网页需要登录后方能访问。用户可通过右侧好友栏添加好友，待目标用户同意后，好友关系建立。

b) 用户可以通过我的上传管理自己私有的视频，通过上边栏的搜索框在私有视频和公开视频中进行搜索。选中某一视频后可以选择创建观影任务，可以在观影之约中分别查看自己发起的和应邀的小组信息。

c) 需要确定时间和共同观影人员，被邀请的成员会直接加入小组中，可以在观影之约中分别查看自己发起的和应邀的小组信息。

d) 小组的创建者可以修改观影时间和人员，参与者可选择退出观影。

3. 任务管理

a) 选中某一视频后即可以选择创建观影任务， 需要确定时间和共同观影人员，被邀请的成员会直接加入小组中，可以在观影之约中分别查看自己发起的和应邀的小组信息。

b) 小组的创建者可以修改观影时间和人员，参与者可选择退出观影。

c) 点击某一小组的标题进入视频播放页面。

4. 视频上传以及管理

用户可以上传视频到公共视频库中，并且能够修改自己上传的视频的名称以及标签。

5. 视频关键字搜索

提供视频关键字搜索功能，用户在我们的网站上可以使用关键字搜索想要观看的视频。

6. 文字、弹幕交流

文字聊天部分位于视频播放界面右侧。使用 socket.io 模块实现。

7. 实时语音交流

在同步观影时，用户可通过语音实时进行交流，最大限度地模拟在电影院中一起看电影的场景。

8. 实时视频交流（待定）

在同步观影时，用户可边观影边进行视频聊天，该功能由于实现起来较为困难，故在此设为待定功能。

三、技术细节

1. API

经过调研 ,目前国内视频网站中仅乐视视频与土豆视频提供对播放进行控制的 API，两者中又以乐视的 API 较为完善，故我们将优先使用乐视 API。下面列出我们将重点使用的乐视 API：

a) seekTo：跳转到视频的某一时间点进行播放

b) getVideoTime：获得视频的当前播放时间

c) video.list：根据视频名称模糊搜索，返回符合的视频列表

d) video.upload.init：视频上传初始化

e) video.upload.resume：视频断点续传

f) video.update：编辑视频名称以及标签等信息

g) video.get：获取单个视频的信息

具体见：

http://help.lecloud.com/Wiki.jsp?page=PC4.0

http://help.lecloud.com/Wiki.jsp?page=VideoInfo

2. 数据库组织

我们采用 mongodb 数据库进行存储。

1) 用户信息数据库

以用户 id 为主键，包含密码、邮箱、好友、个人视频、所属观影小组等信息。

2) 观影任务数据库

以任务的执行时间（即用户设定的观影时间）排序， 存储任务的建立时间、执行时间、影片名称、影片在第三方网站（如乐视）的 url、影片的播放地址 url

等信息。通过轮询的方式获得最近待执行的任务。

3. 同步观影的实现

根据用户设置的开始播放时间与当前时间的差值，使用 seekTo 进行播放进度跳转。由于乐视 API 存在一些 bug，不能精确地跳转到设置的秒数，这里采用了多次跳转直到符合要求的方法将同步观影的时间误差尽量限制在 2 秒以内，但偶尔会出现时间误差较大的情况。

4. 上传以及管理视频的实现

调用 video.upload.init 进行视频上传初始化，调用 video.upload.resume 进行断点续传，上传完毕后通过 video.update 编辑视频名称等信息，编辑完毕后通过 video.get 获得视频信息。

5. 视频关键字搜索的实现

通过调用 video.list 这一 API 进行视频检索。

6. 网站前端实现

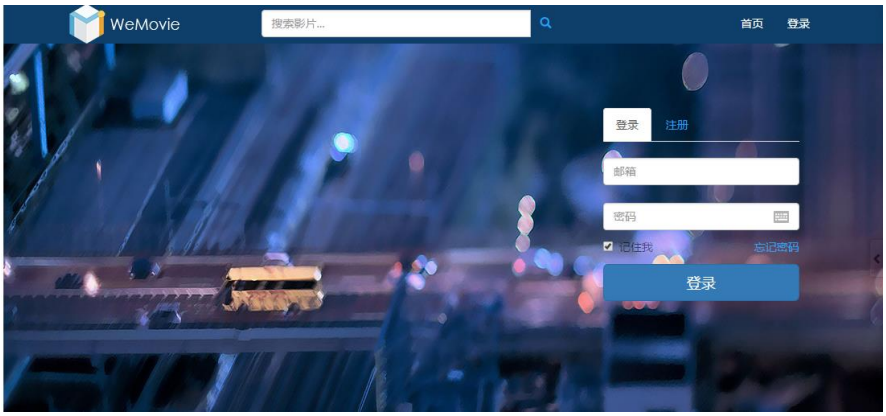
前端以 Bootstrap + jQuery 为基础框架，后端使用 node.js。

Bootstrap 是基于 HTML、CSS、JAVASCRIPT 的，它简洁灵活，使得 Web 开发更加快捷。目前，网站采用扁平化设计，清新简约风格，共分为 3 个页面，具体功能及交互如下：

1) 主页：

a) 顶部提供搜索栏，观影页以及登录选项；搜索栏可供搜索网站已有电影，点击搜索按钮会跳转至搜索结果页；点击影片按钮会跳转至当前观影页（如无当前观影则显示“无当前观影”）；点击登录按钮会弹出登录/注册悬浮窗，登陆后跳转到个人中心页面；

b) 网页主体为背景及登录窗口；

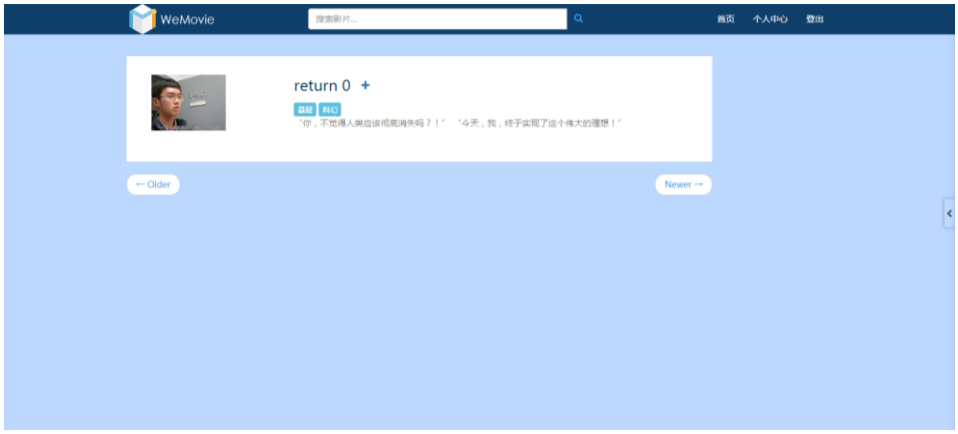


2) 观影页：

a) 顶部状态栏不变；网页左上部分为视频窗口，不提供开始/暂停功能，可提供弹幕开关，音量增减，全屏缩放等功能；网页左下部分为打字区以供发送弹幕；右上方为视频 or 音频区；右下方显示参与观影人员；

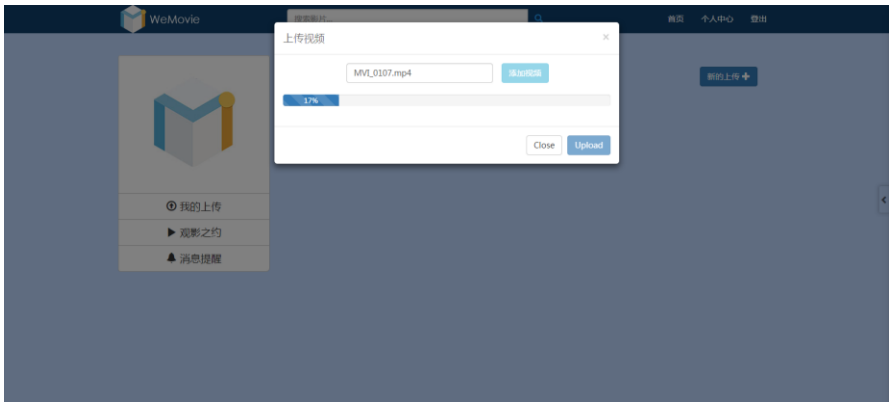
b) 弹幕实现可采用 jQuery 插件，类似于 Bilibili 等视频弹幕网站上的弹幕功能。彩色弹幕、顶端底端弹幕、自定义弹幕速度、实时调整透明度等弹幕该具备的基本功能，都可提供；

3) 搜索页：显示搜索结果，点击结果中的电影条目则可跳转到该电影的详情页；



4) 个人中心：显示我的上传，我的观影记录以及消息提醒，同时提供好友栏，在好友栏中点击搜索按钮即可添加好友

a) 我的上传页面中提供上传按钮，点击即可进行上；对于已上传且已审核通过的视频可点击发起观影并邀请好友一起观影。



b) 观影之约页面显示我发起的观影与我收到的观影邀请，可删除记录

四、运行流程

1. 网站启动

a) 首先需要启动数据库

b) 使用 npm start 启动网站，因为需要监听 80 端口，可能需要管理员权限

2. 调用关系

a) 首先调用的是 bin/www 文件，作用是启动服务器，开始监听 80 端口，以及聊天系统 socket.io 的初始化

b) bin/www 文件调用 app.js，作用是启动 express 框架，调用 routes/index.js 进行路由。这里包含了对大多数依赖包的调用。

c) routes/index.js 包含了网站大部分用户功能的实现。不同的用户信息在数据库下的存取方式定义于 models/文件夹下。

d) routes/index.js 中的大多数路由返回的是对 ejs 文件的渲染，ejs 文件存放于 views/文件夹下。

3. 会话功能

我们使用会话功能对一些用户和操作信息进行本地存储，从而减少信息的冗余传输。

依赖包 express-session，具体信息存储于变量 req.session 中。

4. 用户系统

主要包含基本的用户信息，好友信息，上传的视频信息，观影小组信息等。

5. 聊天系统

聊天系统在 bin/www 文件中被初始化，确定了服务器对一些信号 signal 的反馈方式。用户向服务器发送聊天信息的模块定义于前端，views/letv.ejs 下。