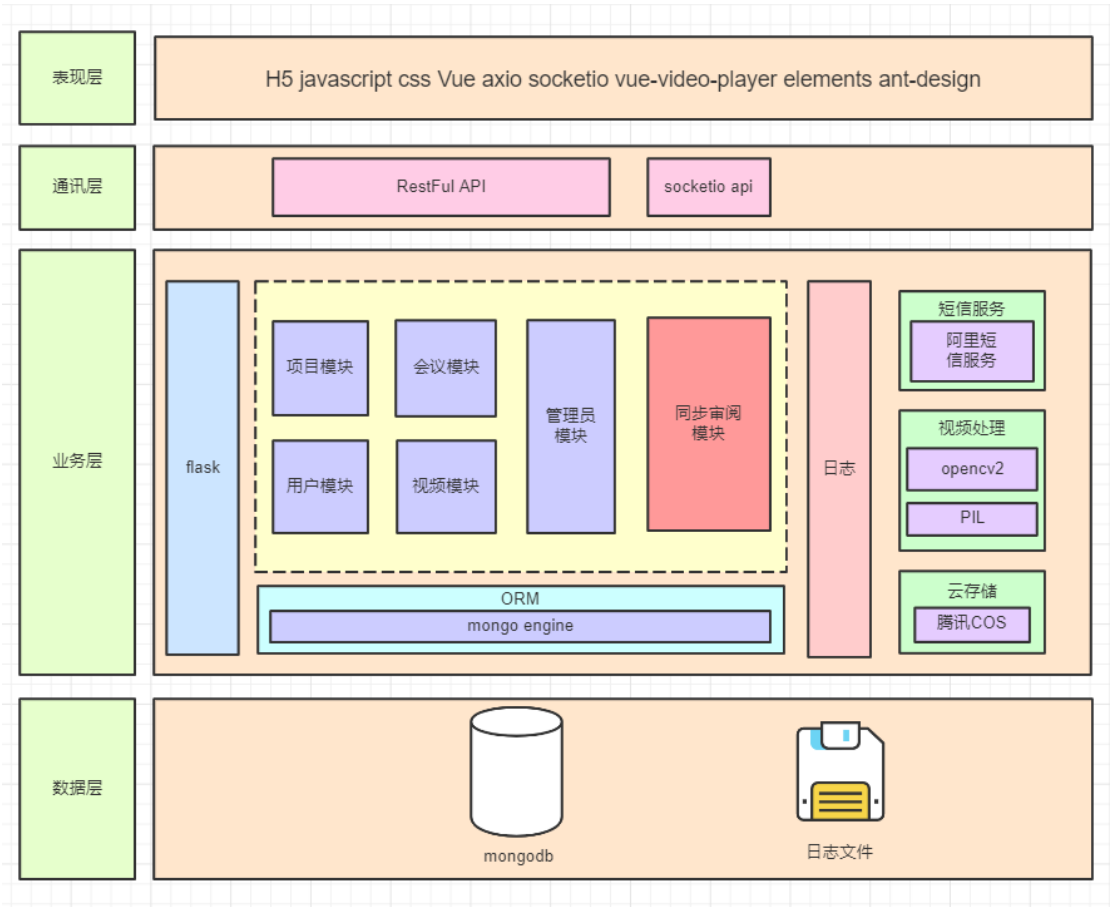# 软件设计文档

题　　目　 视频审阅平台　

学　　院　智能与计算学部　

专　　业　　 软件工程　　　

学　　号　 3018216116　　

姓　　名　　 尤嘉宁　　　

学　　号　 3018216081　　

姓　　名　　 任林杰　　　

学　　号　 3018216084　　

姓　　名　　 田家硕　　　

2020 年　　 12 月　 2 日

目录

目录

一、架构图

表现层	H5 javascript css Vue axio socketio vue-video-player elements ant-design

通讯层	RestFul API　　socketio api

业务层	flask

业务层模块:
- 项目模块
- 会议模块
- 用户模块
- 视频模块
- 管理员模块
- 同步审阅模块

ORM
mongo engine

日志

短信服务
- 阿里短信服务

视频处理
- opencv2
- PIL

云存储
- 腾讯COS

数据层	mongodb　　日志文件

二、数据库设计

用户集合：user

```json
{
    "_id": ObjectId("5fcb47a8872ad7704cb534c4"),
    "username": "用户姓名",
    "password": "pbkdf2:sha256:150000$Mt8GOJ6b$159b6b6ab6d555be22f6841b
258f5f2bebccecaa088b47818e8199578f88ec79",
    "mobileNum": "123456789",
    "mail": "123456@qq.com",
    "avatar": "https://hexo-blog-1123456.cos.ap-
beijing.myqcloud.com/video_review//img/5fcb47a8872ad7704cb534c4/5778565
2139.jpg",
    "company": "tju",
    "hasProject": [
        "5fcb4837872ad7704cb534c5",
        "5fcb4b3a3815cb9e40a1daa1"
    ],
    "uploadVideo": [
        "5fcb486d872ad7704cb534c6",
        "5fcb4f312fd62669086c8da6"
    ],
    "message": [
        {
            "fromId": "5fcb46a9872ad7704cb534c1",
            "fromName": "闹钟你别闹",
            "projectId": "5fcb46f7872ad7704cb534c2",
            "projectName": "希儿",
            "type": 3,
            "messageId": 0,
            "date": 1607354129.7034593,
            "content": {
                "word": "这是我的项目，快来加入吧"
            },
            "hasProcess": 0,
            "hasRead": 1
        }
    ],
    "hasMeeting": [
        "5fd04e7dd7032a3086ba9596"
    ],
    "admin": false,
```

```json
    "alive": true,
    "joinProject": [
        "5fcb46f7872ad7704cb534c2"
    ]
}
```

项目集合：project

```json
{
    "_id": ObjectId("5fcb46f7872ad7704cb534c2"),
    "projectName": "希儿",
    "owner": "5fcb46a9872ad7704cb534c1",
    "waitJoin": [
        "5fcd920e923cbdeb8acb0d3d",
        "5fcb47a8872ad7704cb534c4"
    ],
    "hasVideo": [
        "5fcb4716872ad7704cb534c3",
        "5fcd8cfb923cbdeb8acb0d3c"
    ],
    "hasMeeting": [
        "5fd03d552058891b4785ea30",
        "5fd04d93d7032a3086ba9595"
    ],
    "alive": true,
    "member": [
        {
            "_id": "5fcb93db2fd62669086c8dac",
            "title": "审阅人员"
        }
    ]
}
```

视频集合：video

```json
{
    "_id": ObjectId("5fd06759110e8a17dea75fa7"),
    "videoName": "miku.mp4",
    "owner": "5fcb46a9872ad7704cb534c1",
    "duration": 15,
    "permission": 0,
```

```
    "url": "https://hexo-blog-123456.cos.ap-
beijing.myqcloud.com/video_review/video/5fcb46a9872ad7704cb534c1/934541
64867.mp4",
    "cover": [
        "https://hexo-blog-123456.cos.ap-
beijing.myqcloud.com/video_review/img/5fcb46a9872ad7704cb534c1/93464177
158.jpg",
        "https://hexo-blog-123456.cos.ap-
beijing.myqcloud.com/video_review/img/5fcb46a9872ad7704cb534c1/93464665
790.jpg"
    ],
    "belongTo": "5fd0661e110e8a17dea75fa6",
    "comment": [],
    "password": "",
    "description": "",
    "hasReview": 0,
    "reviewResult": 0,
    "reviewSummary": "",
    "createDate": 1607487321.5010457,
    "alive": true
}
```

会议集合：meeting

```
{
    "_id": ObjectId("5fd03d552058891b4785ea30"),
    "title": "这是会议标题",
    "ownerId": "5fcb46a9872ad7704cb534c1",
    "ownerName": "闹钟你别闹",
    "belongTo": "5fcb46f7872ad7704cb534c2",
    "startTime": 1607482708000,
    "endTime": 1607482709000,
    "meetingUrl": "todo",
    "txMeetingId": "todo",
    "note": "这里是会议备注",
    "alive": true
}
```

三、前后端接口设计

参见《接口设计文档》

四、UI 设计

参见《UI 设计.psd》

五、概要设计
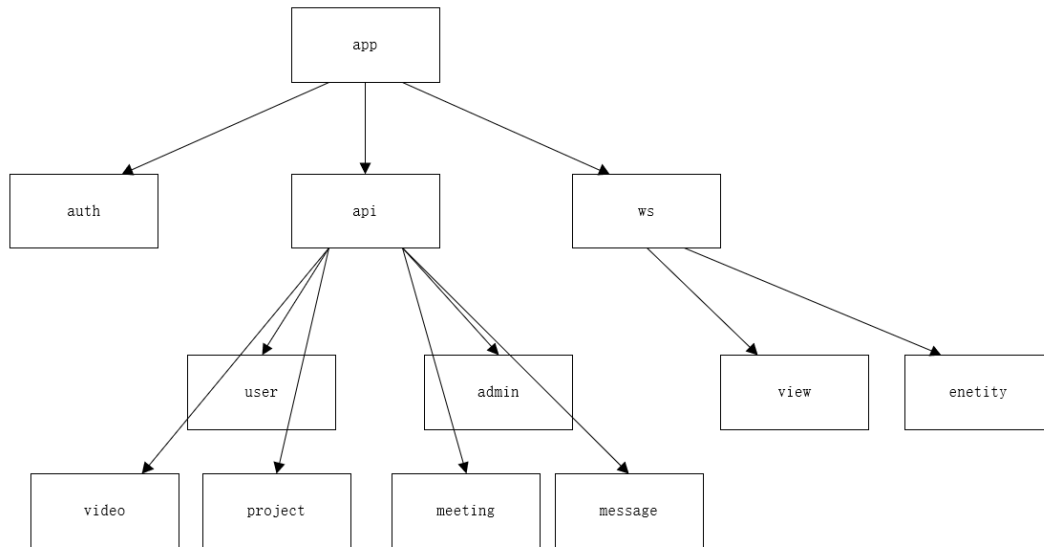
上图表示本项目后端的目录结构，

- app：为项目总体模块。
- auth：鉴权模块，管理用户登录后的 token 分发，刷新，鉴权。
- api：RestFul 接口模块，对外提供所有 HTTP Restful 接口。
- user：用户模块，管理用户相关 Restful 接口。
- admin：管理员模块，提供管理员专用 Restful 接口
- video：视频模块，提供视频相关 Restful 接口。
- project：项目模块，提供视频审阅平台中用户所创建项目相关的所有接口。
- meeting：会议模块，提供平台中会议相关接口。
- message：消息模块，提供平台中用户所需的通知的相关接口。
- ws：Socket.io 接口模块，提供同步审阅相关的接口。
- view：同步审阅业务模块，实现了各个 socket.io 接口的业务流程部分。
- entity：同步审阅实体模块，定义了同步审阅所需要的各个实体类。

六、类设计

SidManager
+ members: Dict[str, MeetingMember]
+ rooms: Dict[str, MeetingRoom]
+ enter_meetingroom(sid, meeting_id, user_id): MeetingMember
+ destroy_meetingroom(meeting_id):void
+ get_meetingRoom_by_meetingid(meeting_id: str):MeetingRoom
+ disconnect_sid(sid:str): void

MeetingRoom
+ member_list:Dict[str, MeetingMember]
+ player: VideoPlayer
+ meeting:Meeting
+ project:Project
+ meeting_id:str
+ manager_id:str
+ command_cache:Tuple
+ time_lock: float
+ get_member_list(): Dict
+ add_member(user_id:str, member:Meeting[Member]):void
+ delete_member(user_id):void
+ get_comment_list():void
+ add_comment():void
+ remove_comment(user_id:str, comment_id:str):void
+ push_cache(is_player:bool, position:float, url:str, type:int):void
+ is_echo(is_player, position:float, url:str, type:int):void
+ guess_states(type:int, video_id:str, position:float, comment_id:str)

VideoPlayer
+ video: Video
+ is_play: bool
+ position: float
+ play(position:float):void
+ pause(position): void
+ move_process(postion:float):void
+ change_video(video_id:str):void
+ get_video_info():Dict

MeetingMember
+ user: User
+ sid:str
+ control: bool
+ comment: bool
+ room: MeetingRoom
+ get_member_item(): List
+ can_control():bool
+ is_manager():bool
+ pause_video(position:float, **kwargs):void
+ start_comment(position:float, **kwargs):void
+ move_video_position(position:float, **kwargs):void
+ play_video(position:float, **kwargs):void
+ end_comment(position:float, **kwargs):void
+ change_video_src(video_id:str, **kwargs):void
+ move_to_comment(position:float, **kwargs):void
+ do_operation(type:int, **kwargs):void

Meeting
+ title: str
+ ownerId: str
+ ownerName: str
+ belongTo: str
+ startTime: float
+ endTime: float
+ meetingUrl: str
+ talkMeetingId: str
+ get_meeting_by_id(meeting_id, deep:bool):Meeting
+ get_meeting_by_projectid(project_id, deep:bool):List[Meeting]
+ get_meeting_by_ownerid(user_id:str, deep:bool):List[Meeting]
+ delete_meeting():void

Video
+ videoName: str
+ owner: str
+ duration: int
+ permission: int
+ url: str
+ cover: List[str]
+ belongTo: str
+ comment: List[Comment]
+ password: str
+ description: str
+ hasReview: int
+ reviewResult: int
+ reviewSummary: str
+ createDate: float
+ alive: bool
+ get_video_by_id(video_id, deep:bool=False): Video
+ delete_video(): void
+ set_review_result(result:bool, summary:str): void

Comment
+ commentId: int
+ fromId: str
+ fromName: str
+ position: int
+ image: str
+ content: str
+ date: float
+ alive: bool

User
+ userName: str
+ password: str
+ mobileNum: str
+ mail: str
+ avatar: str
+ company: str
+ hasProject: List[str]
+ joinProject: List[str]
+ uploadVideo: List[str]
+ message: List[Message]
+ hasMeeting: List[str]
+ admin: bool
+ alive: bool
+ has_user(userId:str, deep:bool):bool
+ get_user_by_id(user_id:str, deep:bool):User
+ get_user_by_mobileNum(mobileNum:str, deep:bool): User
+ receive_message(message: Message):void
+ get_message_by_id(message_id:int):void
+ read_message(message_id:int):void
+ process_message(message_id:int):void

Project
+ projectName: str
+ owner: str
+ member: List[ProjectMember]
+ waitJoin: List[str]
+ hasVideo: List[str]
+ hasMeeting: List[str]
+ alive: bool
+ has_project(project_id:str, deep:bool):bool
+ get_project_by_id(project_id:str, deep:bool):Project
+ wait_to_user(user_id:str): void
+ confirm_to_join(user:User, is_agree:bool): void
+ remove_video(video_id:str):void
+ remove_member(user:User, is_leave:bool, is_owner):void
+ dissolution():void
+ member_id_list()->List[str]

ProjectMember
+ userId: str
+ title: str

Message
+ fromId: str
+ fromName: str
+ projectId: str
+ projectName: str
+ type: int
+ messageId: int
+ date: float
+ content: MessageContent
+ hasProcess: int
+ hasRead: int
+ fill_content(**kwargs):void

MessageContent
+ videoName: str
+ reviewResult: str
+ meetingId: str
+ word: str
+ processResult: bool
+ upload_new_video_message(video_name:str): MessageContent
+ video_has_review(video_name:str, review_result:bool, **kwargs): MessageContent
+ book_new_meeting(meeting_id:str, **kwargs):MessageContent
+ invite_join_project(word:str, **kwargs):MessageContent
+ invite_has_process(process_result:bool, **kwargs):MessageContent
+ remove_project_member(**kwargs):MessageContent