

闲亭

软件设计说明书



学 院 智能与计算学部

专 业 软件工程

学 号 3018216108

姓 名 满显凡

学 号 3018216072

姓 名 李青颖

学 号 3018216078

姓 名 毛文瑞

学 号 3018218052

姓 名 白嘉欣

学 号 3018216127

姓 名 鲍子龙

目录

一、	API 设计	4
二、	数据库设计	5
2.1	E/R 图.....	5
三、	结构设计	6
3.1	软件体系结构图.....	6
3.2	类设计图.....	7
四、	类设计	14
4.1	annotation 包.....	14
4.2	aspect 包.....	14
4.3	config 包	15
4.4	Exception 包.....	15
4.5	filter 包.....	16
4.6	security 包.....	16
4.7	util 包	17
4.8	controller 包.....	19
4.9	service 包	27
4.10	mapper 包.....	35
4.11	bo 包.....	51
4.12	vo 包.....	55
4.13	entity 包	61
4.14	global 包.....	65
五、	UI 设计	67
5.1	用户前台 UI.....	67
5.1.1	起始页:.....	67
5.1.2	登录/注册:.....	67
5.1.3	文章列表界面:	68
5.1.4	文章详情界面:	68
5.1.5	手绘列表界面:	69
5.1.6	手绘详情界面:	69
5.1.7	搜索:	70
5.1.8	个人主页:	70
5.1.9	信息修改界面:	71
5.1.10	发布界面:	71
5.1.11	主题设置界面:	72
5.2	管理员后台 UI.....	74
5.2.1	管理员登录界面.....	74
5.2.2	管理员管理.....	74
5.2.3	添加管理员.....	75
5.2.4	用户管理.....	75
5.2.5	文章管理.....	76
5.2.6	查看文章详情	76
5.2.7	手绘管理.....	77

5.2.8 查看手绘详情	77
5.2.9 分类管理	78
5.2.10 添加分类	78
5.2.11 主题管理	79
5.2.12 添加主题	79
5.2.13 功能管理	80
5.2.14 日志管理	80

一、API 设计

见 API 接口文档

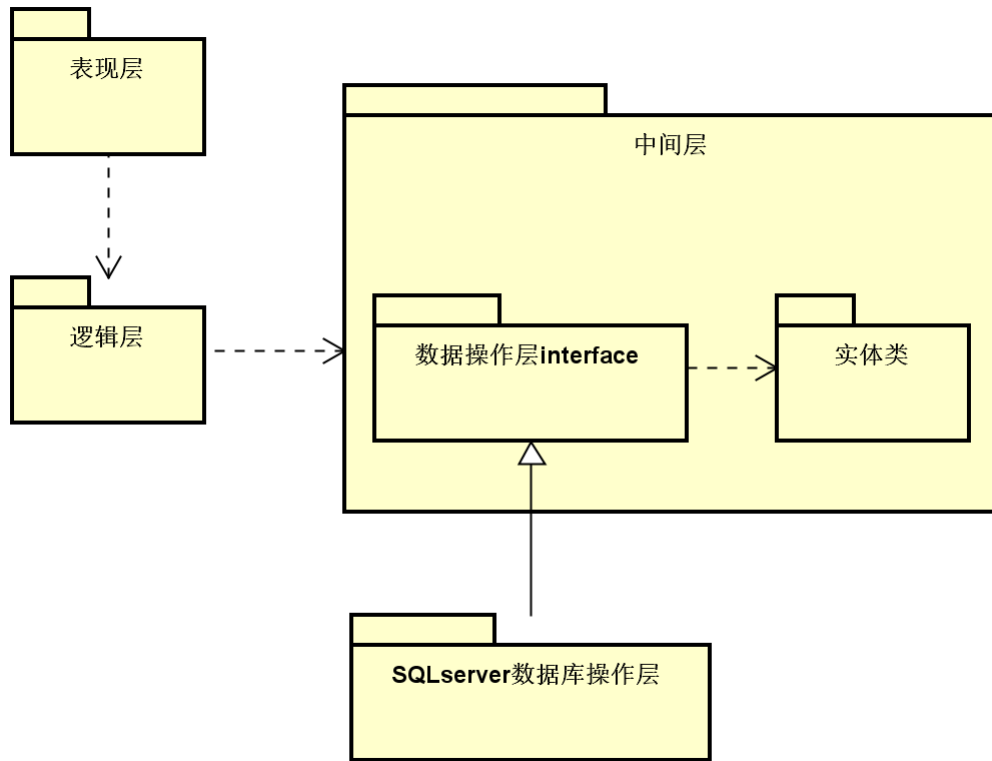
二、数据库设计

2.1 E/R 图



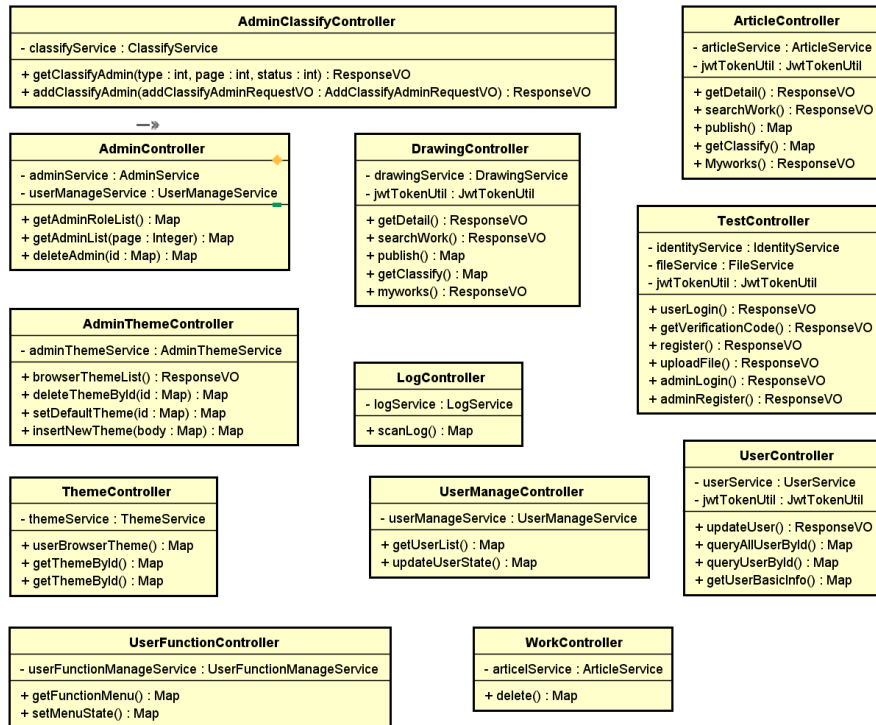
三、结构设计

3.1 软件体系结构图

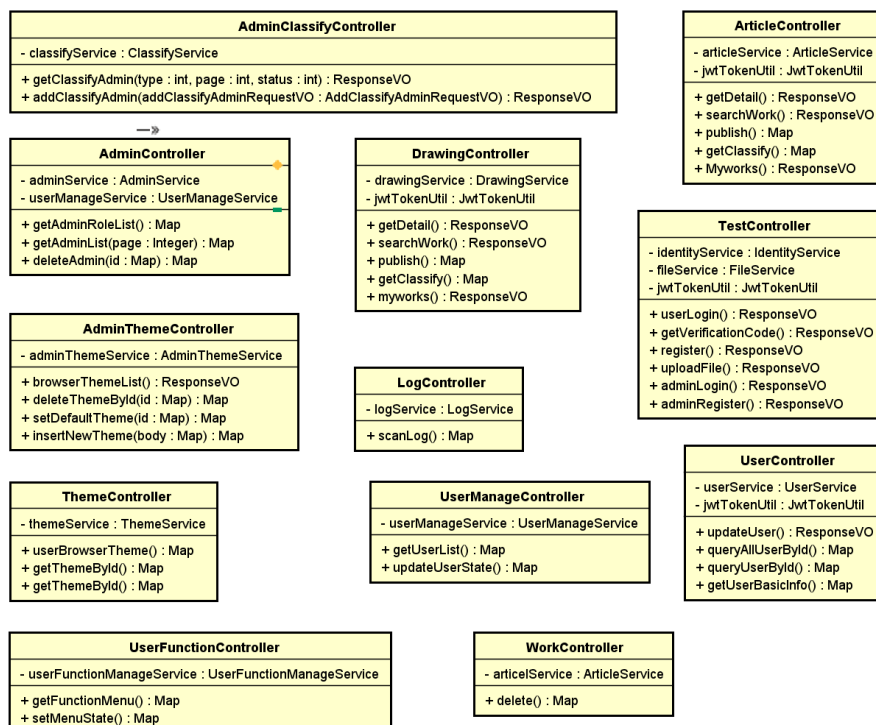


3.2 类设计图

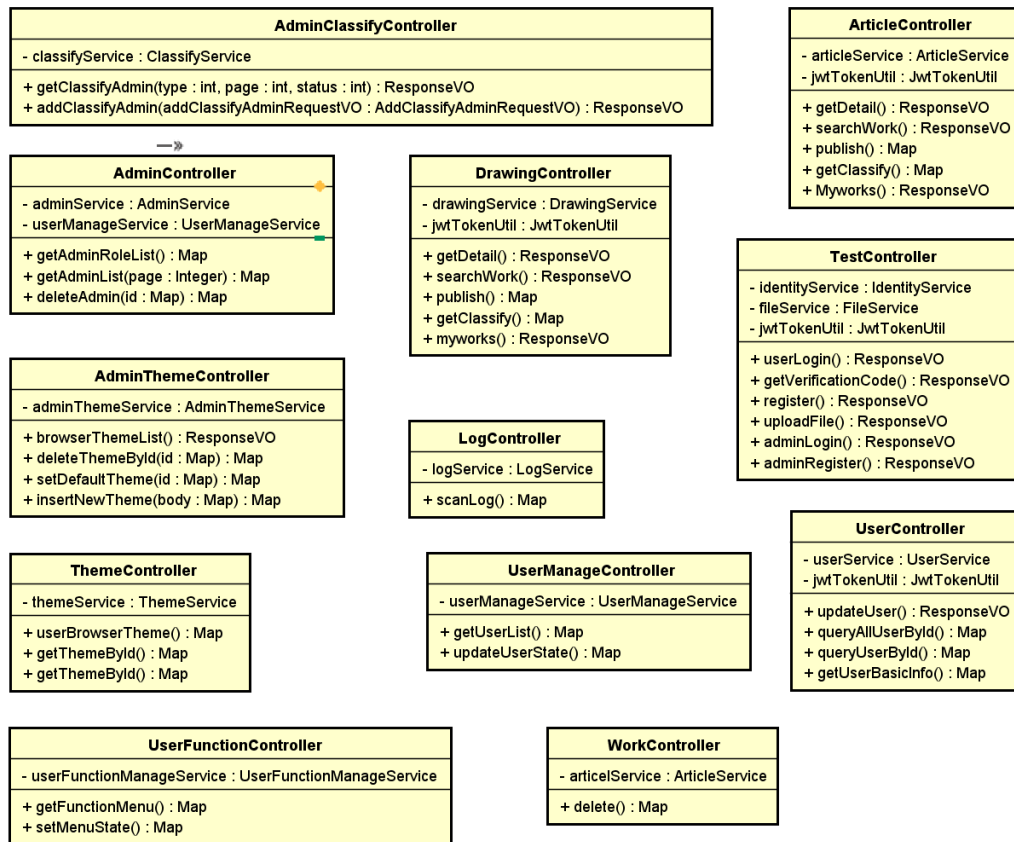
3.2.1 controller



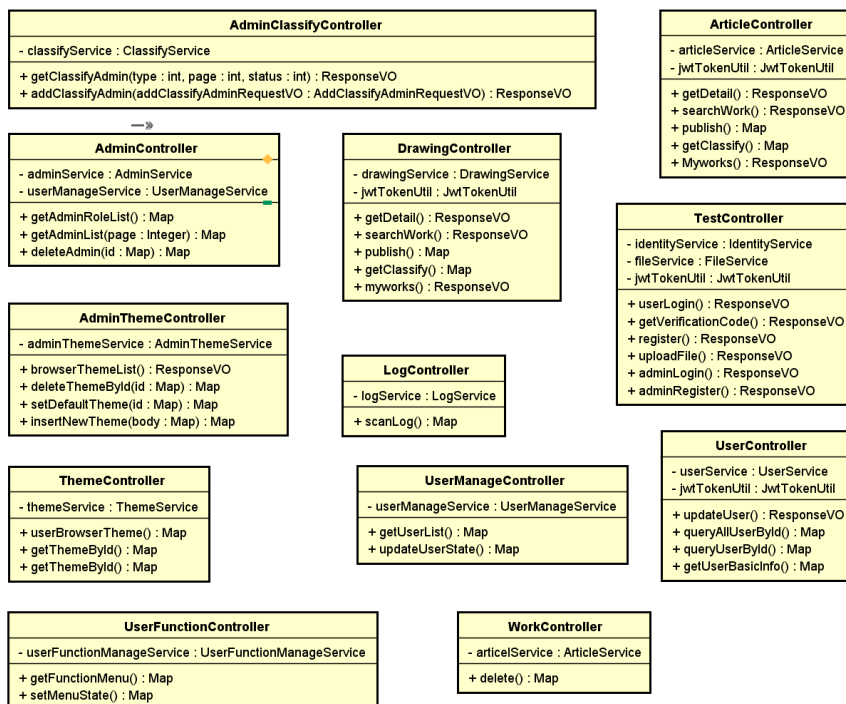
3.2.2 service



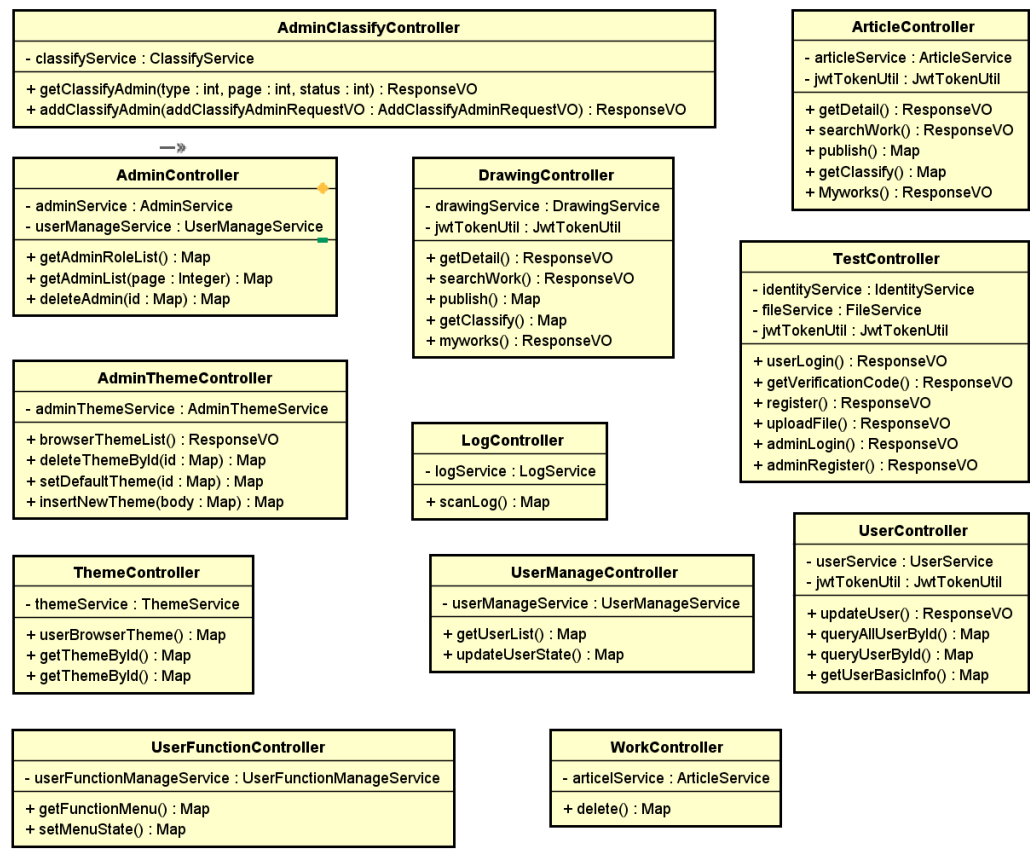
3.2.3 mapper



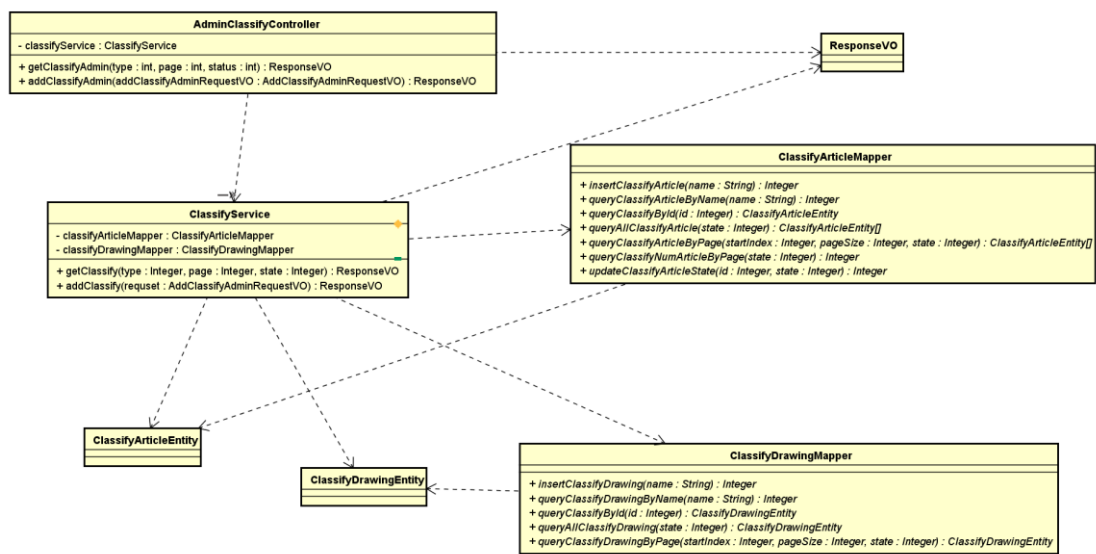
3.2.4 entity

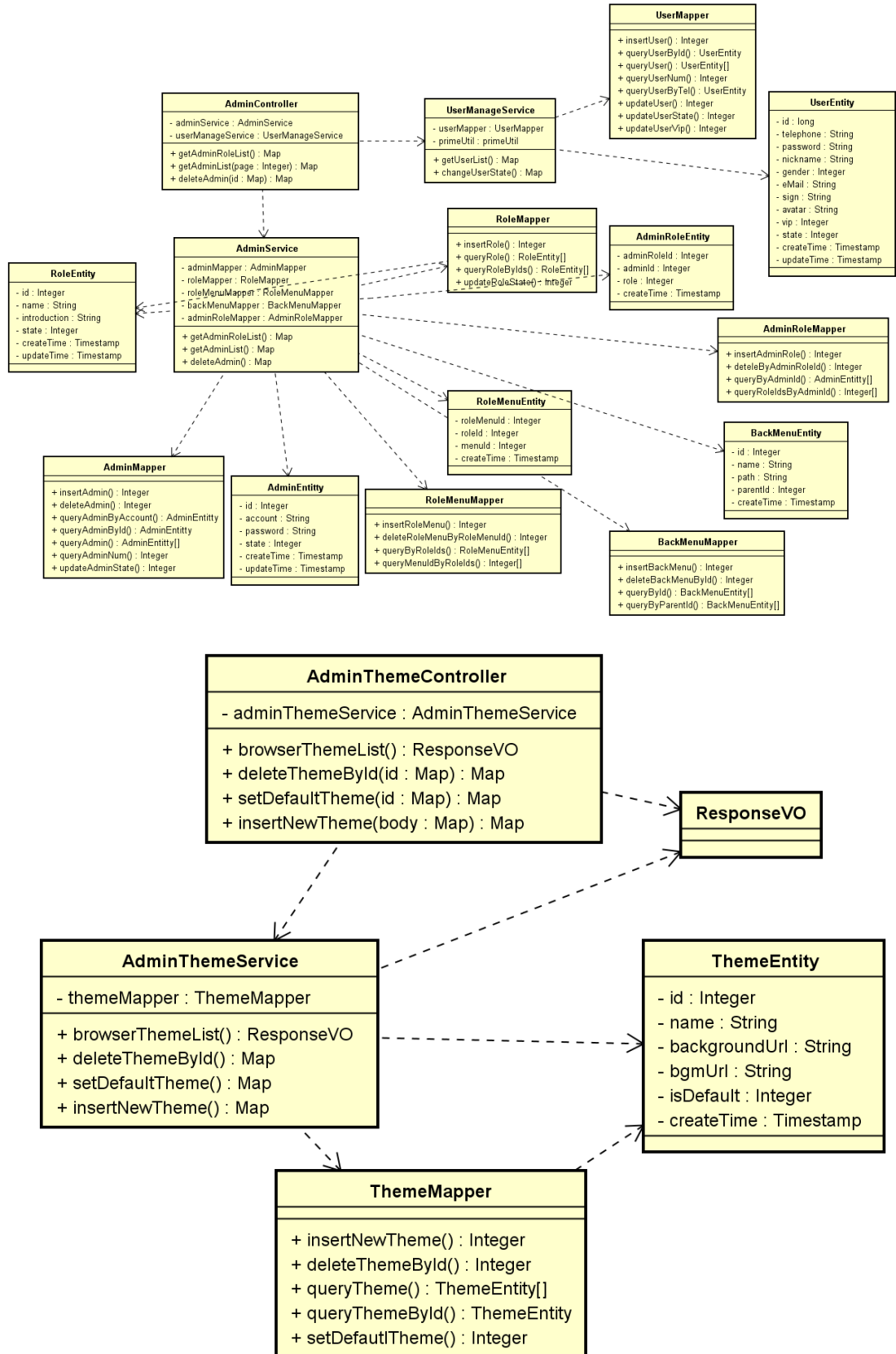


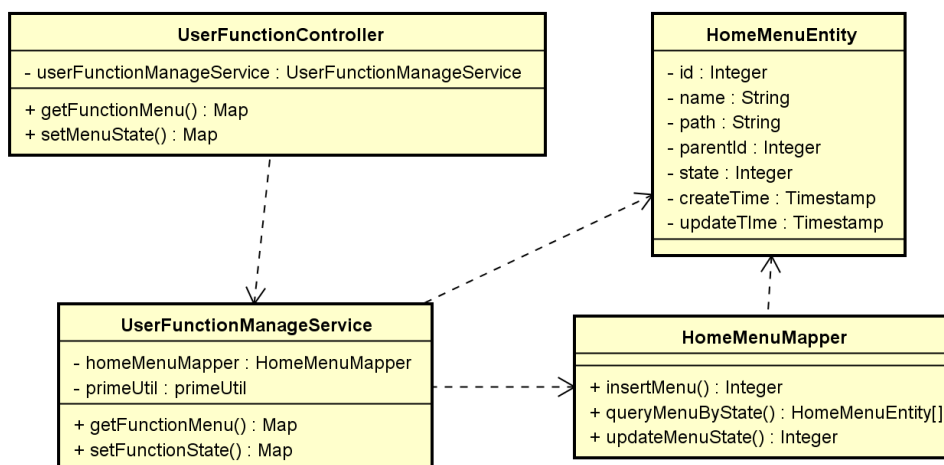
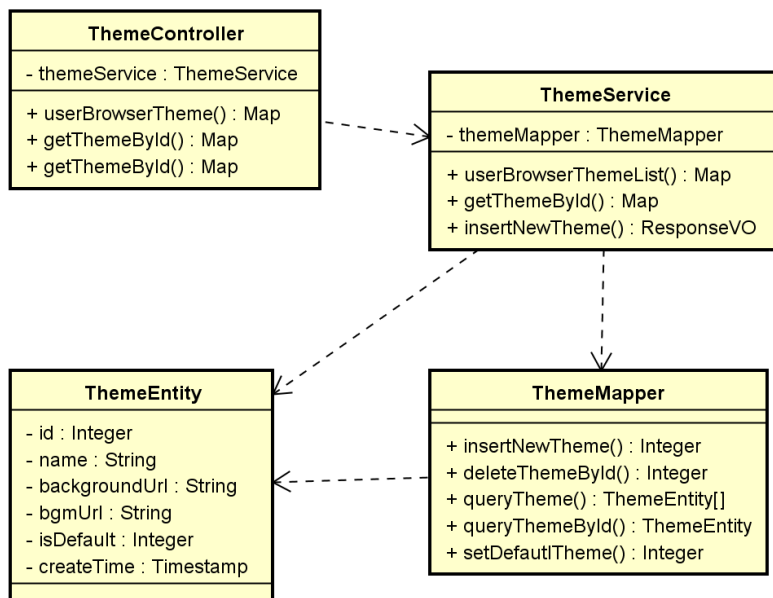
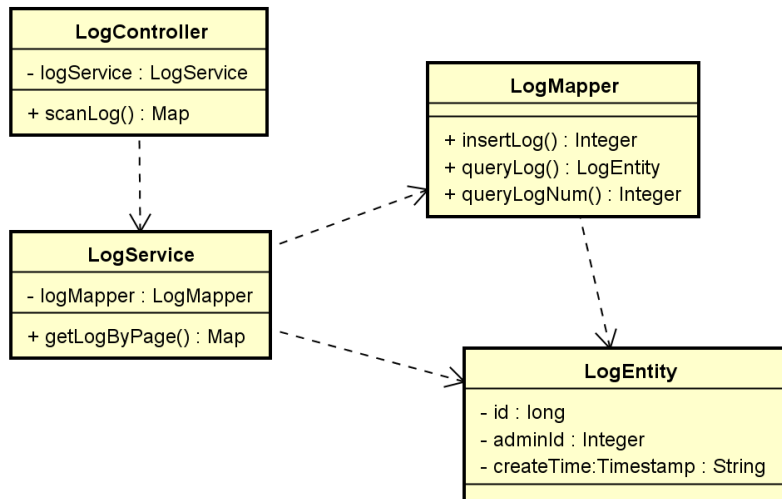
3.2.5 vo

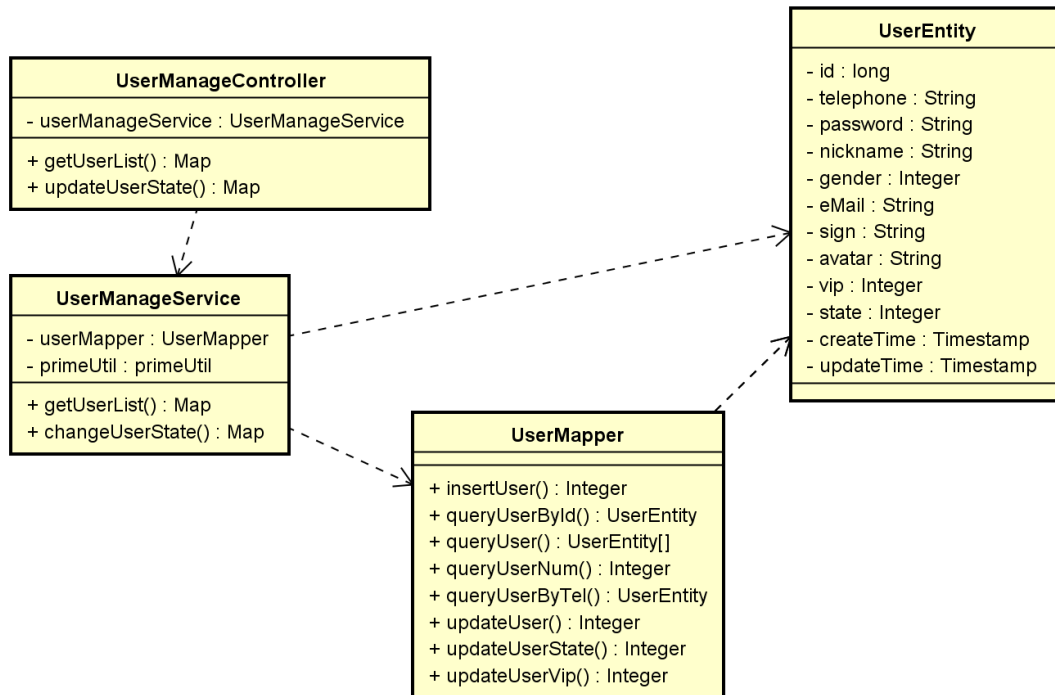
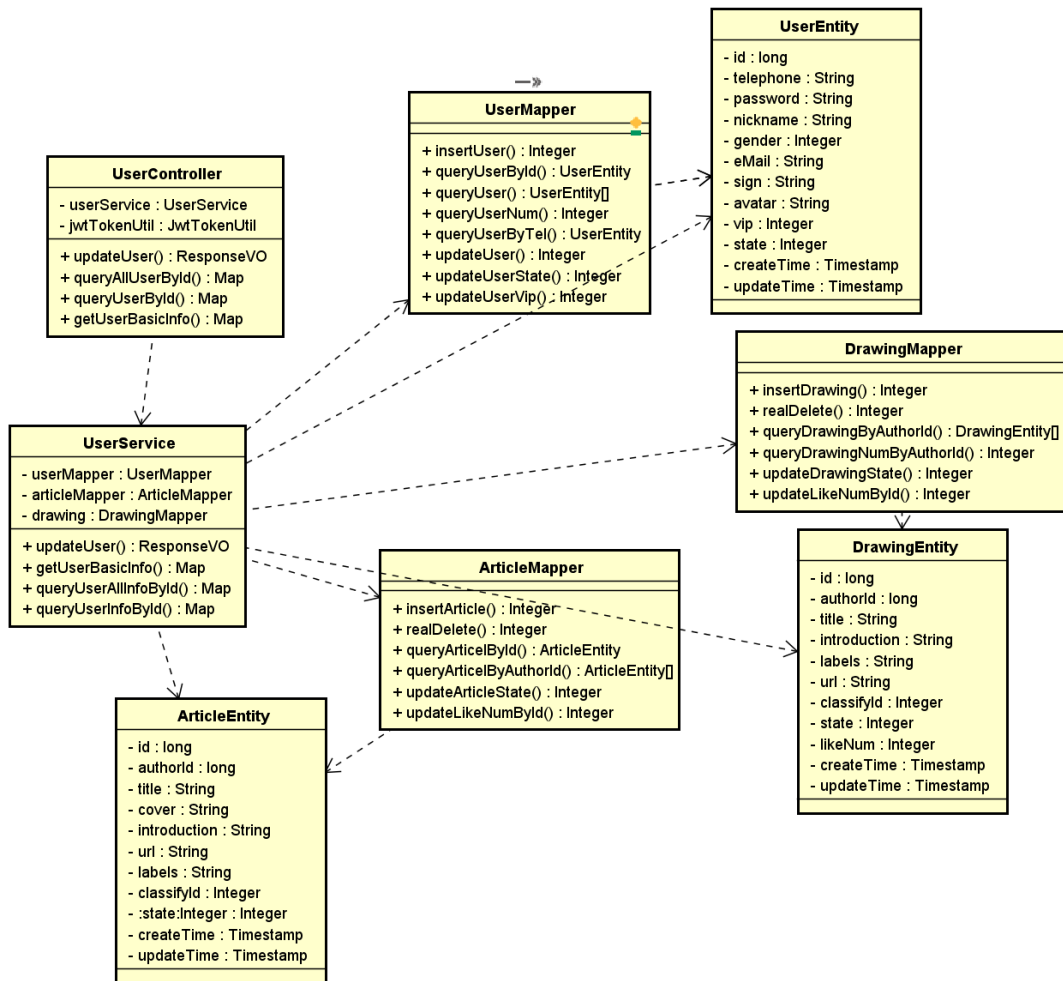


3.2.6 类关系









四、类设计

4.1 annotation 包

4.1.1 OperationLogAnnotation

```
1. public @interface OperationLogAnnotation {  
2.     String operName() default ""; // 方法名  
3. }
```

4.2 aspect 包

4.2.1 OperationLogAspect

```
1. public class OperationLogAspect {  
2.  
3.     @Autowired  
4.     LogMapper logMapper;  
5.  
6.     @Autowired  
7.     JwtTokenUtil jwtTokenUtil;  
8.  
9.     /**  
10.    * 设置操作日志切入点 在注解的位置切入代码  
11.    */  
12.    @Pointcut("@annotation(ap.deepstroll.annotation.OperationLogAnnotation)"  
13.    )  
14.  
15.  
16.    /**  
17.    * 写入日志  
18.    * @param joinPoint  
19.    * @param result  
20.    * @throws Throwable  
21.    */  
22.    @AfterReturning(returning /**  
23.    * 记录操作日志  
24.    * @param joinPoint 方法的执行点  
25.    * @param result 方法返回值  
26.    * @throws Throwable  
27.    */ = "result", value = "operLogPoinCut()")
```

```

28.     public void saveOperLog(JoinPoint joinPoint, Object result) throws Throwable {}
29.
30. }

```

4.3 config 包

4.3.1 WebSecurityConfig

```

1. public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
2.
3.     @Autowired
4.     public void configureAuthentication(AuthenticationManagerBuilder authenticationManagerBuilder) throws Exception {}
5.
6.     @Bean
7.     public PasswordEncoder passwordEncoder() {
8.         return new BCryptPasswordEncoder();
9.     }
10.
11.    @Bean
12.    public JwtAuthenticationTokenFilter authenticationTokenFilterBean() throws Exception {
13.        return new JwtAuthenticationTokenFilter();
14.    }
15.
16.    // 控制访问
17.    @Override
18.    protected void configure(HttpSecurity httpSecurity) throws Exception { }
19.
20. }

```

4.4 Exception 包

4.4.1 StateException

```

1. public class StateException extends Throwable {
2.

```

```

3.     private String message;
4.     public StateException(String key,String val){
5.         this.message = "required key "+ key +" has no value of "+ val;
6.     }
7.     public String getMessage()
8. }

```

4.5 filter 包

4.5.1 JwtAuthenticationTokenFilter

```

1. public class JwtAuthenticationTokenFilter extends OncePerRequestFilter {
2.
3.     private UserDetailsService userDetailsService;
4.
5.     private UserDetailsService adminDetailsService;
6.
7.     private JwtTokenUtil jwtTokenUtil;
8.
9.     private String tokenHeader;
10.
11.    private String tokenHead;
12.    // 依据 token 判断用户身份 权限
13.    @Override
14.    protected void doFilterInternal(
15.        HttpServletRequest request,
16.        HttpServletResponse response,
17.        FilterChain chain
18.    ) throws ServletException, IOException {}
19.
20. }

```

4.6 security 包

4.6.1 AdminDetailsServiceImp

```

1. public class AdminDetailsServiceImp implements UserDetailsService {
2.
3.     AdminMapper adminMapper;
4.
5.     AdminRoleMapper adminRoleMapper;
6.
7.     RoleMapper roleMapper;

```



```

8.
9.     //返回管理员对应的 UserDetails
10.    @Override
11.    public UserDetails loadUserByUsername(String id) throws UsernameNotFoundException {
12.
13. }

```

4.6.2 UserDetailsServiceImpl

```

1. public class UserDetailsServiceImpl implements UserDetailsService {
2.
3.     UserMapper userMapper;
4.
5.     返回用户对应的 UserDetails
6.     @Override
7.     public UserDetails loadUserByUsername(String id) throws UsernameNotFoundException {
8.
9. }

```

4.6.3 UserDetailsImp

```

1. public class UserDetailsImp implements UserDetails {
2.
3.     private static final long serialVersionUID = 1L;
4.     // 用户 id
5.     private final Long id;
6.     // 用于登录 user 为 telephone admin 为 account
7.     private final String username;
8.     private final String password;
9.     // 用户状态
10.    private final Integer state;
11.    private Collection<? extends GrantedAuthority> authorities;
12.
13. }

```

4.7 util 包

4.7.1 DateFormatUtil

```

1. public class DateFormatUtil {
2.
3.     private static DateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm");

```

```
4.  
5. public static String getDate(Timestamp t);  
6. }
```

4.7.2 HttpRequestUtil

```
1. @Component  
2. public class HttpRequestUtil {  
3.  
4.     protected String sendHttpRequest(String url, String param);  
5. }
```

4.7.3 JwtTokenUtil

```
1. @Component  
2. public class JwtTokenUtil{  
3.  
4.     private static final String CLAIM_KEY_ID = "id";  
5.     private static final String CLAIM_KEY_ROLE = "role";  
6.  
7.     @Value("${jwt.secret}")  
8.     private String secret;  
9.  
10.    private Claims getClaimsFromToken(String token)  
11.  
12.    public String getIdFromToken(String token);  
13.  
14.    public String[] getIdAndRoleFromToken(String token);  
15.  
16.    public String generateToken(String id, String role);  
17.  
18.    private String generateToken(Map<String, Object> claims);  
19. }
```

4.7.4 SMSMessageUtil

```
1. @Component  
2. public class SMSMessageUtil extends HttpRequestUtil {  
3.  
4.     public Boolean sendSMSMessage()  
5.  
6. }
```

4.7.5 primeUtil

```

1. public class primeUtil {
2.
3. /**
4.  * 检查是不是手机号
5.  * @param phone
6.  * @return
7.  */
8. public Boolean checkPhone(String phone);
9. /
10. **
11.  * 判断某类型是否存在
12.  * @param i
13.  * @param type
14.  * @return
15.  */
16. public Boolean checkvalue(Integer i ,String type);

```

4.8 controller 包

4.8.1 AdminClassifyController

```

1. public class AdminClassifyController {
2.
3. @GetMapping("/api/admin/classify")
4. public ResponseVO getClassifyAdmin(
5.     @RequestParam Integer type,
6.     @RequestParam Integer page,
7.     @RequestParam(required = false) Integer status);
8.
9. @PostMapping("/api/admin/classify/add")
10. public ResponseVO addClassifyAdmin(@RequestBody AddClassifyAdminRequestV
    0 addClassifyAdminVO);
11. }

```

4.8.2 AdminController

```

1. public class AdminController {
2. /**
3.  * 获取管理员角色测试通过
4.  * mwr dei
5.  * @return
6.  */
7. @GetMapping("/api/admin/role")

```

```

8.     @PreAuthorize("hasAnyRole('chiefAdmin','rightAdim')")
9.     public Map<String,Object> getAdminRoleList();
10.
11.    /**
12.     * 获取管理员列表
13.     * mwr dei
14.     * @param
15.     * @return
16.     */
17.    @GetMapping("/api/admin/list")
18.    @PreAuthorize("hasAnyRole('chiefAdmin','rightAdim')")
19.    public Map<String,Object>getAdminList(@RequestParam Integer page);
20.
21.    /**
22.     * 删除管理员，实际更改管理员状态
23.     * mwr dei
24.     * @param id
25.     * @return
26.     */
27.    @PostMapping("/api/admin/delete")
28.    @PreAuthorize("hasAnyRole('chiefAdmin','rightAdim')")
29.    @OperationLogAnnotation(operName = "删除管理员")
30.    public Map<String,Result> deleteAdmin(@RequestBody Map<String,Integer> i
        d);
31. }

```

4.8.3 AdminThemeController

```

1. public class AdminThemeController {
2.
3.    /**
4.     * 获取主题列表
5.     * @return
6.     */
7.    @GetMapping("/api/admin/theme")
8.    @PreAuthorize("hasAnyRole('chiefAdmin','themeAdmin')")
9.    public ResponseVO browserThemeList();
10.
11.    /**
12.     * 删除指定主题
13.     * @param id
14.     * @return
15.     */
16.    @PostMapping("/api/admin/theme/delete")

```

```

17.     @PreAuthorize("hasAnyRole('chiefAdmin','themeAdmin')")
18.     @OperationLogAnnotation(operName = "删除主题")
19.     public Map<String, Result> deleteThemeById(@RequestBody Map<String,Integer> id);
20.
21.     /**
22.      * 修改默认主题
23.      * @param id
24.      * @return
25.      */
26.     @PostMapping("/api/admin/theme/default")
27.     @PreAuthorize("hasAnyRole('chiefAdmin','themeAdmin')")
28.     public Map<String, Result> setDefaultTheme(@RequestBody Map<String,Integer> id);
29.
30.     /**
31.      * 管理员添加主题
32.      * @param
33.      * @return
34.      */
35.     @PostMapping("/api/admin/theme/add")
36.     @PreAuthorize("hasAnyRole('chiefAdmin','themeAdmin')")
37.     @OperationLogAnnotation(operName = "添加主题")
38.     public Map<String, Result> insertNewTheme(@RequestBody Map<String, String> body);
39. }

```

4.8.4 ArticleController

```

1. public class ArticleController {
2.
3.     //获取详情
4.     @GetMapping("/api/article/detail")
5.     @PreAuthorize("hasAnyRole('common','chiefAdmin','workAdmin')")
6.     public ResponseVO getDetail(@RequestParam Long id);
7.
8.     //分类浏览
9.     @GetMapping("/api/article/search")
10.    @PreAuthorize("hasAnyRole('common','chiefAdmin','workAdmin')")
11.    public ResponseVO searchWork(@RequestParam Integer page,
12.    @RequestParam(required = false) String key,
13.    @RequestParam(required = false) Integer classify,
14.    @RequestParam(required = false) Integer status);
15.

```

```

16.    //发布
17.    @PostMapping("/api/person/article/add")
18.    @PreAuthorize("hasAnyRole('common')")
19.    public Map Publish(@RequestBody ArticleVO req, @RequestHeader HttpHeaders headers);
20.
21.    @GetMapping("/api/article/classify")
22.    @PreAuthorize("hasAnyRole('common','chiefAdmin','workAdmin')")
23.    public Map<String,Object> getClassify();
24.
25.    @GetMapping("/api/person/article")
26.    @PreAuthorize("hasRole('common')")
27.    public ResponseVO myWorks(
28.        @RequestParam(required = false) String key,
29.        @RequestParam(required = false) Integer classify,
30.        @RequestParam Integer page,
31.        @RequestHeader HttpHeaders headers);
32. }

```

4.8.5 DrawingController

```

1. public class DrawingController {
2.
3.    //获取详情
4.    @GetMapping("/api/drawing/detail")
5.    @PreAuthorize("hasAnyRole('common','chiefAdmin','workAdmin')")
6.    public ResponseVO getDetail(@RequestParam Long id);
7.
8.    //分类浏览
9.    @GetMapping("/api/drawing/search")
10.   @PreAuthorize("hasAnyRole('common','chiefAdmin','workAdmin')")
11.   public ResponseVO searchWork(@RequestParam Integer page,
12.       @RequestParam(required = false) String key,
13.       @RequestParam(required = false) Integer classify,
14.       @RequestParam(required = false) Integer status);
15.
16.   @PostMapping("/api/person/drawing/add")
17.   @PreAuthorize("hasRole('common')")
18.   public Map Publish(@RequestBody DrawingVO req, @RequestHeader HttpHeaders headers);
19.
20.   @GetMapping("/api/drawing/classify")
21.   @PreAuthorize("hasAnyRole('common','chiefAdmin','workAdmin')")
22.   public Map<String,Object> getClassify();
23.

```

```

24.     @GetMapping("/api/person/drawing")
25.     @PreAuthorize("hasRole('common')")
26.     public ResponseVO myworks(
27.         @RequestParam(required = false) String key,
28.         @RequestParam(required = false) Integer classify,
29.         @RequestParam Integer page,
30.         @RequestHeader HttpHeaders headers);
31. }

```

4.8.6 LogController

```

1. public class LogController {
2.     @GetMapping("/api/admin/log")
3.     @PreAuthorize("hasAnyRole('chiefAdmin','logAdmin')")
4.     public Map<String,Object> scanLog(@RequestParam Integer page);
5. }

```

4.8.7 TestController

```

1. public class TestController {
2.     @PostMapping("/api/login")
3.     public ResponseVO userLogin(@RequestBody UserLoginRequestVO userLoginRequestVO);
4.
5.     @PostMapping("/api/ssm")
6.     public ResponseVO getVerificationCode(@RequestBody GetVerificationCodeRequestVO getVerificationCodeRequestVO);
7.
8.     @PostMapping("/api/register")
9.     public ResponseVO register(@RequestBody RegisterRequestVO registerRequestVO);
10.
11.     @PostMapping("/api/file")
12.     @PreAuthorize("hasAnyRole('common','chiefAdmin','themeAdmin')")
13.     public ResponseVO uploadFile(@RequestParam MultipartFile file);
14.
15.     @PostMapping("/api/admin/login")
16.     public ResponseVO adminLogin(@RequestBody AdminLoginRequestVO adminLoginRequestVO);
17.
18.     @PostMapping("/api/admin/add")
19.     @PreAuthorize("hasAnyRole('chiefAdmin','rightAdmin')")
20.     @OperationLogAnnotation(operName = "添加管理员")
21.     public ResponseVO adminRegister(@RequestBody AdminRegisterRequestVO adminRegisterRequestVO);

```

```
22.  
23. }
```

4.8.8 ThemeController

```
1. public class ThemeController {  
2.  
3.     /**  
4.      * user 浏览内置主题  
5.      * 如果是默认主题的话, 就显示背景和音乐, 否则只显示 id 和 name  
6.      * @return  
7.      */  
8.     @GetMapping("/api/config/themeList")  
9.     @PreAuthorize("hasAnyRole('common')")  
10.    public Map<String, Object> userBrowserThemeList();  
11.  
12.    /**  
13.     * id 获取主题(背景音乐、背景图片)  
14.     * @param id  
15.     * @return  
16.     */  
17.    @GetMapping("/api/config/theme")  
18.    @PreAuthorize("hasAnyRole('common')")  
19.    public Map<String, Object> getThemeById(@RequestParam Integer id);  
20.  
21. }
```

4.8.9 UserController

```
1. public class UserController {  
2.  
3.     /**  
4.      * 修改用户信息  
5.      * @return  
6.      */  
7.     @PostMapping("/api/person/updateInfo")  
8.     @PreAuthorize("hasRole('common')")  
9.     public Result updateUser(@RequestBody UserVo req, @RequestHeader HttpHeaders headers);  
10.  
11.     //根据 id 获得用户全部信息  
12.     @GetMapping("/api/person/totalInfo")  
13.     @PreAuthorize("hasRole('common')")  
14.     public Map<String, Object> queryAllUserById(@RequestHeader HttpHeaders headers);
```



```

15.
16.     //根据 id 获得用户信息
17.     @GetMapping("/api/person/info")
18.     @PreAuthorize("hasRole('common')")
19.     public Map<String, Object> queryUserById(@RequestHeader HttpHeaders headers);
20.
21.     /**
22.      * 获取用户基本信息
23.      * @param
24.      * @return
25.      */
26.     @GetMapping("/api/person/basic")
27.     @PreAuthorize("hasRole('common')")
28.     public Map<String, Object> getUserBasicInfo(@RequestHeader HttpHeaders headers);
29. }

```

4.8.10 UserFunctionManagerController

```

1. public class UserFunctionManagerController {
2.
3.     /**
4.      *功能菜单列表
5.      * mwr
6.      * dei
7.      * @return
8.      */
9.     @GetMapping("/api/admin/menu")
10.    @PreAuthorize("hasAnyRole('chiefAdmin','userFunctionAdmin')")
11.    public Map<String, Object> getFunctionMenu();
12.
13.    /**
14.     * 设置功能状态
15.     * @param req
16.     * mwr dei
17.     * @return
18.     */
19.    @PostMapping("/api/admin/menu/status")
20.    @PreAuthorize("hasAnyRole('chiefAdmin','userFunctionAdmin')")
21.    @OperationLogAnnotation(operName = "设置功能状态")
22.    public Map<String, Result> setMenuState(@RequestBody Map<String, Integer> req);
23. }

```

4.8.11 UserManagerController

```
1. public class UserManagerController {
2.
3.     /**
4.      * 获取用户列表, 包含条件搜索
5.      * mwr dei
6.      * @param req
7.      * @return
8.      */
9.     @GetMapping("/api/admin/user")
10.    @PreAuthorize("hasAnyRole('chiefAdmin','userAdmin')")
11.    public Map<String,Object> getUsersList(
12.        @RequestParam(required = false) String key,
13.        @RequestParam(required = false) Integer status,
14.        @RequestParam(required = false) Integer vip,
15.        @RequestParam Integer page);
16.
17.    /**
18.     * 更新用户状态
19.     * mwr dei
20.     * @param req
21.     * @return
22.     */
23.    @PostMapping("/api/admin/user/status")
24.    @PreAuthorize("hasAnyRole('chiefAdmin','userAdmin')")
25.    @OperationLogAnnotation(operName = "更改用户状态")
26.    public Map<String, Result>updateUserState(@RequestBody Map<String,Object>
27.        > req);
28. }
```

4.8.12 WorkController

```
1. public class WorkController {
2.
3.     @PostMapping("/api/person/works/delete")
4.     @PreAuthorize("hasRole('common')")
5.     public Map<String,Result> delete(@RequestBody DeleteWorkRequestVO request, @RequestHeader HttpHeaders headers);
6. }
```

4.9 service 包

4.9.1 AdminService

```
1. @Service("adminService")
2. public class AdminService {
3.
4.     /**
5.      * 获取管理员角色
6.      * mwr dei
7.      * @param
8.      * @return */
9.     public Map<String,Object> getAdminRoleList();
10.
11.    /**
12.     * 获取管理员列表
13.     * mwr
14.     * @param page
15.     * @return
16.     */
17.    public Map<String,Object> getAdminList(Integer page);
18.
19.    /**
20.     * 修改管理员状态
21.     * mwr dei
22.     * @param
23.     * @return
24.     */
25.    public Map<String,Result> deleteAdmin(Map<String,Integer> req);
26. }
```

4.9.2 AdminThemeService

```
1. @Service("adminThemeService")
2. public class AdminThemeService {
3.
4.     /**
5.      * 获取主题列表
6.      * @return
7.      */
8.     public ResponseVO browserThemeList();
9.
10.    /**
11.     * 删除指定主题
```

```

12.     * @param id
13.     * @return
14.     */
15.     public Map<String ,Result> deleteThemeById(Integer id);
16.
17.     /**
18.      * 修改默认主题
19.      * @param id
20.      * @return
21.      */
22.     public Map<String, Result> setDefaultTheme(Integer id);
23.
24.     /**
25.      * 管理员添加主题
26.      * @param
27.      * @return
28.      */
29.     public Map<String, Result> insertNewTheme(String name, String bg, String
        bgm);
30. }

```

4.9.3 ArticleService

```

1. @Service("articleService")
2. @Primary
3. public class ArticleService extends WorkService {
4.
5.     @Override
6.     public ResponseVO searchWork(String title,String labels,Integer classify
        Id,Integer state,Integer page);
7.
8.     /**
9.      * lqy
10.     * 获取作品的分类
11.     * @return
12.     */
13.     @Override
14.     public ResponseVO getDetail(Long id);
15.
16.     public ResponseVO myWorks(Long id, String title, String labels, Integer
        classifyId, Integer page);
17.
18.     /**
19.      * 获取作品的分类
20.      * @return

```

```

21.     */
22.     @Override
23.     public Map<String, Object> getClassify();
24.
25.     @Override
26.     public Map<String, Result> Publish(WorkVO work, long id);
27.
28.     public Result Collection(Map<String, String> req);
29.
30.     /**
31.      * 删除作品
32.      * @param id
33.      * @param type
34.      * @return
35.      */
36.     public Map<String, Result> deleteWork(Long id, Integer type, Long author
        Id);
37. }

```

4.9.4 ClassifyService

```

1. @Service
2. public class ClassifyService {
3.
4.     public ResponseVO getClassify(Integer type, Integer page, Integer state)
        ;
5.
6.     public ResponseVO addClassify(AddClassifyAdminRequestVO request);
7. }

```

4.9.5 DrawingService

```

1. @Service("drawingService")
2. public class DrawingService extends WorkService {
3.
4.     private String transNull(String s);
5.
6.     @Override
7.     public ResponseVO searchWork(String title,String labels,Integer classify
        Id,Integer state,Integer page);
8.
9.     /**
10.      * 获得分类
11.      * @return
12.      */

```

```

13.     @Override
14.     public ResponseVO getDetail(Long id);
15.
16.     public ResponseVO myWorks(Long id, String title, String labels, Integer
        classifyId, Integer page);
17.
18.     /**
19.      * 获得分类
20.      * @return
21.      */
22.     @Override
23.     public Map<String, Object> getClassify();
24.
25.     @Override
26.     public Map<String, Result> Publish(WorkVO work, long id);
27.
28.     public Result Collection(Map<String, String> req);
29.
30.     public Map<String, Result> deleteWork(Long id, Integer type, Long author
        Id);
31. }

```

4.9.6 FileService

```

1. @Service
2. public class FileService {
3.
4.     public ResponseVO uploadFile(MultipartFile mFile);
5.
6.     private String getKey();
7.
8.     private String upload(File file);
9.
10.    private File transMultipartFileToFile(MultipartFile mFile) throws IOExce
        ption;
11. }

```

4.9.7 IdentityService

```

1. @Service
2. public class IdentityService {
3.
4.     /**
5.      * 用户登录 返回 token id
6.      * @author mxf

```

```
7.      * @return
8.      */
9.      public ResponseVO userLogin(UserLogInRequestVO userLogInRequestVO);
10.
11.     /**
12.      * 为指定手机号发送验证码
13.      * @author mxf
14.      * @param
15.      * @return
16.      */
17.     public ResponseVO getVerificationCode(GetVerificationCodeRequestVO getVerificationCodeRequestVO);
18.
19.     /**
20.      * 绑定手机号 注册新用户
21.      * @author mxf
22.      * @param nickname
23.      * @param pwd
24.      * @param telephone
25.      * @param code
26.      * @return
27.      */
28.     public ResponseVO bindTelephone(RegisterRequestVO registerRequestVO);
29.
30.     /**
31.      * 管理员登录 返回 token
32.      * @author mxf
33.      * @param account
34.      * @param password
35.      * @return
36.      */
37.     public ResponseVO adminLogin(AdminLogInRequestVO adminLogInRequestVO);
38.
39.     /**
40.      * 添加管理员
41.      * @author mxf
42.      * @param account
43.      * @param password
44.      * @param roleId
45.      * @return
46.      */
47.     public ResponseVO adminRegister(AdminRegisterRequestVO adminRegisterRequestVO);
48. }
```

4.9.8 LogService

```
1. @Service("logService")
2. public class LogService {
3.
4.     /**
5.      *
6.      * 查看日志
7.      * mwr
8.      * @param req
9.      * @return
10.     */
11.     public Map<String, Object> getLogByPage(Integer page);
12. }
```

4.9.9 SMSService

```
1. @Service
2. public class SMSService {
3.
4.     public String sendSMS(String telephone, String code);
5. }
```

4.9.10 ThemeService

```
1. @Service("themeService")
2. public class ThemeService {
3.     /**
4.      * user 浏览内置主题
5.      * 如果是默认主题的话, 就显示背景和音乐, 否则只显示 id 和 name
6.      * @return
7.      */
8.     public Map<String, Object> userBrowserThemeList();
9.
10.    /**
11.     * id 获取主题(背景音乐、背景图片)
12.     * @param id
13.     * @return
14.     */
15.     public Map<String, Object> getThemeById(Integer id);
16.
17.    /**
18.     * 用户添加主题
19.     * @param themeEntity
20.     * @return

```



```

21.     */
22.     public Result insertNewTheme(ThemeEntity themeEntity);
23.
24. }

```

4.9.11 UserFunctionManageService

```

1. @Service("userFunctionManageService")
2. public class UserFunctionManageService {
3.
4.     /**
5.      * 获取功能菜单列表
6.      * mwr dei
7.      * @return
8.      */
9.     public Map<String,Object>getFunctionMenu();
10.
11.    /**
12.     * 设置功能状态
13.     * mwr dei
14.     * @param req
15.     * @return
16.     */
17.     public Map<String, Result> setFunctionState(Map<String,Integer>req);
18. }

```

4.9.12 UserManageService

```

1. @Service("userManageService")
2. public class UserManageService {
3.
4.     /**
5.      * 获取用户列表
6.      * mwr dei
7.      * @return
8.      */
9.     public Map<String,Object> getUserList(String key,Integer status,Integer
vip,Integer page);
10.
11.    /**
12.     * 更改用户状态 status vip
13.     * mwr dei
14.     * @param req
15.     * @return
16.     */

```

```
17.     public Map<String,Result> changeUserState(Map<String,Object> req);
18.
19. }
```

4.9.13 UserService

```
1. @Service("userService")
2. public class UserService {
3.
4.     /**
5.      * 修改用户信息
6.      * @author lqy
7.      * @param
8.      * @return
9.      */
10.    public Result updateUser(UserVo req,long id);
11.
12.    /**
13.     * 获取用户基本信息
14.     * nickname & avatar vip
15.     * @param
16.     * @return
17.     */
18.    public Map<String,Object> getUserBasicInfo(Long id);
19.
20.    /**
21.     * 返回指定 id 用户信息:
22.     * @param id
23.     * @return
24.     */
25.    public Map<String,Object> queryUserInfoById(Long id);
26.
27. }
```

4.9.14 VIPService

```
1. @Service("VIPService")
2. public class VIPService extends UserService {
3.
4.     public Map getHotDrawing();
5.
6.     public Map getHotArticle();
7. }
```

4.9.15 WorkService

```

1. @Service("workService")
2. public abstract class WorkService {
3.
4.     /**
5.      * 分类浏览用户分享的作品
6.      * @param
7.      * @return
8.      */
9.
10.    public abstract ResponseVO searchWork(String title,String labels,Integer
        classifyId,Integer state,Integer page);
11.
12.    /**
13.     * 获取作品的分类
14.     * @return
15.     */
16.    public Map<String,Object> getClassify(){return null;};
17.
18.    /**
19.     * 获取作品详情
20.     */
21.    public abstract ResponseVO getDetail(Long id);
22.
23.    /**
24.     * 发布文章
25.     * @param
26.     * @return
27.     */
28.    public abstract Map<String,Result> Publish(WorkVO work,long id);
29.
30. }

```

4.10 mapper 包

4.10.1 AdminMapper

```

1. @Mapper
2. @Repository
3. public interface AdminMapper {
4.
5.     /**
6.      * 插入管理员信息
7.      * @param adminEntity
8.      * @return

```

```
9.      */
10.     public Integer insertAdmin(AdminEntity adminEntity);
11.
12.     public Integer deleteAdmin(@Param("id")Integer id);
13.
14.     /**
15.      * 通过账户查询管理员
16.      * @param account
17.      * @return
18.      */
19.     public AdminEntity queryAdminByAccount(@Param("account")String account);
20.
21.     /**
22.      * 通过 id 查询管理员
23.      * @param id
24.      * @return
25.      */
26.     public AdminEntity queryAdminById(@Param("id")Integer id);
27.
28.     /**
29.      * 查询全部管理员
30.      * @author mxf
31.      * @param startIndex
32.      * @param pageSize
33.      * @return
34.      */
35.     public List<AdminEntity> queryAdmin(
36.         @Param("state")Integer state,
37.         @Param("startIndex")Integer startIndex,
38.         @Param("pageSize")Integer pageSize
39.     );
40.
41.     /**
42.      * 查询全部管理员数目
43.      * @return
44.      */
45.     public Integer queryAdminNum(@Param("state")Integer state);
46.
47.     /**
48.      * 修改管理员状态
49.      * @param id
50.      * @param state
51.      * @return
```

```

52.     */
53.     public Integer updateAdminState(@Param("id")Integer id, @Param("state")Integer state);
54.
55. }

```

4.10.2 AdminRoleMapper

```

1. @Mapper
2. @Repository
3. public interface AdminRoleMapper {
4.
5.     /**
6.      * 插入一个管理员以及其角色
7.      * @param adminId
8.      * @param roleId
9.      * @return
10.     */
11.     public Integer insertAdminRole(
12.         @Param("adminId")Integer adminId,
13.         @Param("roleIds")Integer[] roleId
14.     );
15.
16.     /**
17.      * 删除指定 admin role 对应
18.      * @author mx
19.      * @param adminRoleId
20.      * @return
21.     */
22.     public Integer deleteByAdminRoleId(@Param("adminRoleId")Integer adminRoleId);
23.
24.     /**
25.      * 查询某管理员的角色
26.      * @param adminId
27.      * @return
28.     */
29.     public List<AdminRoleEntity> queryByAdminId(@Param("adminId")Integer adminId);
30.
31.     /**
32.      * 查询某管理员的角色 id
33.      * @param adminId
34.      * @return
35.     */

```

```

36.     public List<Integer> queryRoleIdsByAdminId(@Param("adminId")Integer adminId);
37.
38. }
39.

```

4.10.3 ArticleMapper

```

1. @Mapper
2. @Repository
3. public interface ArticleMapper {
4.
5.     /**
6.      * 插入一条新的文章
7.      * @param articleEntity
8.      */
9.     public Integer insertArticle(ArticleEntity articleEntity);
10.
11.     public Integer realDelete(@Param("id")Long id);
12.
13.     /**
14.      * 查询指定 id 的文章信息
15.      * @param id
16.      * @return
17.      */
18.     public ArticleEntity queryArticleById(@Param("id")Long id);
19.
20.     /**
21.      * 查询指定作者的作品
22.      * @param id
23.      * @return
24.      */
25.     public List<ArticleEntity> queryArticleByAuthorId(
26.         @Param("authorId")Long id,
27.         @Param("title")String title,
28.         @Param("label")String label,
29.         @Param("classifyId")Integer classifyId,
30.         @Param("state")Integer state,
31.         @Param("startIndex")Integer startIndex,
32.         @Param("pageSize")Integer pageSize
33.     );
34.
35.     public Integer queryArticleNumByAuthorId(
36.         @Param("authorId")Long id,
37.         @Param("title")String title,

```

```
38.         @Param("label")String label,
39.         @Param("classifyId")Integer classifyId,
40.         @Param("state")Integer state
41.     );
42.
43.     /**
44.      * 根据限定条件搜索文章
45.      * @param title
46.      * @param label
47.      * @param classifyId
48.      * @param state
49.      * @param startIndex notnull
50.      * @param pageSize notnull
51.      * @return
52.      */
53.     public List<ArticleEntity> queryArticleByTitleLabClassifyState(
54.         @Param("title")String title,
55.         @Param("label")String label,
56.         @Param("classifyId")Integer classifyId,
57.         @Param("state")Integer state,
58.         @Param("startIndex")Integer startIndex,
59.         @Param("pageSize")Integer pageSize,
60.         @Param("likeNum")Integer likeNum
61.     );
62.
63.     /**
64.      * 返回符合条件的总数
65.      * @param title
66.      * @param label
67.      * @param classifyId
68.      * @param state
69.      * @param likeNum
70.      * @return
71.      */
72.     public Integer queryArticleNumByTitleLabClassifyState(
73.         @Param("title")String title,
74.         @Param("label")String label,
75.         @Param("classifyId")Integer classifyId,
76.         @Param("state")Integer state,
77.         @Param("likeNum")Integer likeNum
78.     );
79.
80.     /**
81.      * 更新文章状态
```

```

82.     * @param id
83.     * @param state
84.     */
85.     public Integer updateArticleState(@Param("id")Long id, @Param("state")Integer state);
86.
87.     /**
88.     * 增加点赞数 1
89.     * @param id
90.     * @return
91.     */
92.     public Integer updateLikeNumById(@Param("id")Long id, @Param("num")Integer num);
93.
94. }

```

4.10.4 BackMenuMapper

```

1. @Mapper
2. @Repository
3. public interface BackMenuMapper {
4.
5.     /**
6.     * 插入菜单
7.     * @param backMenuEntity
8.     * @return
9.     */
10.    public Integer insertBackMenu(BackMenuEntity backMenuEntity);
11.
12.    /**
13.    * 删除指定菜单
14.    * @param id
15.    * @return
16.    */
17.    public Integer deleteBackMenuById(@Param("id")Integer id);
18.
19.    /**
20.    * 查询一系列指定 id 的菜单信息
21.    * @param ids
22.    * @return
23.    */
24.    public List<BackMenuEntity> queryById(@Param("ids")List<Integer> ids);
25.
26.    /**
27.    * 查询指定爸爸的儿子们

```



```

28.     * @param parentId
29.     * @return
30.     */
31.     public List<BackMenuEntity> queryByParentId(@Param("parentId")Integer pa
        rentId);
32.
33. }

```

4.10.5 ClassifyDrawingMapper

```

1. @Mapper
2. @Repository
3. public interface ClassifyDrawingMapper {
4.
5.     /**
6.      * 插入一个新的分类
7.      * @param name
8.      * @return
9.      */
10.    public Integer insertClassifyDrawing(@Param("name")String name);
11.
12.    public Integer queryClassifyDrawingByName(@Param("name")String name);
13.
14.    public ClassifyDrawingEntity queryClassifyById(@Param("id")Integer id);
15.
16.    /**
17.     * 获取全部分类
18.     * @param state
19.     * @return
20.     */
21.    public List<ClassifyDrawingEntity> queryAllClassifyDrawing(@Param("state
        ")Integer state);
22.
23.    public List<ClassifyDrawingEntity> queryClassifyDrawingByPage(
24.        @Param("startIndex")Integer startIndex,
25.        @Param("pageSize")Integer pageSize,
26.        @Param("state")Integer state
27.    );
28.
29.    public Integer queryClassifyNumDrawingByPage(
30.        @Param("state")Integer state
31.    );
32.

```

```
33.     public Integer updateClassifyDrawingState(@Param("id")Integer id, @Param
    ("state")Integer state);
34. }
```

4.10.6 ClassifyArticleMapper

```
1. @Mapper
2. @Repository
3. public interface ClassifyArticleMapper {
4.
5.     /**
6.      * 添加新的分类
7.      * @param classifyArticleEntity
8.      */
9.     public Integer insertClassifyArticle(@Param("name")String name);
10.
11.     public Integer queryClassifyArticleByName(@Param("name")String name);
12.
13.     public ClassifyArticleEntity queryClassifyById(@Param("id")Integer id);
14.
15.     /**
16.      * 查询分类
17.      * @param state
18.      * @return
19.      */
20.     public List<ClassifyArticleEntity> queryAllClassifyArticle(@Param("state
    ")Integer state);
21.
22.     public List<ClassifyArticleEntity> queryClassifyArticleByPage(
23.         @Param("startIndex")Integer startIndex,
24.         @Param("pageSize")Integer pageSize,
25.         @Param("state")Integer state
26.     );
27.
28.     public Integer queryClassifyNumArticleByPage(
29.         @Param("state")Integer state
30.     );
31.     /**
32.      * 修改分类状态
33.      * @param id
34.      * @param state
35.      */
36.     public Integer updateClassifyArticleState(@Param("id")Integer id, @Param
    ("state")Integer state);
```

```
37.  
38. }
```

4.10.7 DrawingMapper

```
1. @Mapper  
2. @Repository  
3. public interface DrawingMapper {  
4.  
5.     /**  
6.      * 插入一条手绘  
7.      * @param drawingEntity  
8.      * @return  
9.      */  
10.    public Integer insertDrawing(DrawingEntity drawingEntity);  
11.  
12.    public Integer realDelete(@Param("id")Long id);  
13.  
14.    public List<DrawingEntity> queryDrawingByAuthorId(  
15.        @Param("authorId")Long id,  
16.        @Param("title")String title,  
17.        @Param("label")String label,  
18.        @Param("classifyId")Integer classifyId,  
19.        @Param("state")Integer state,  
20.        @Param("startIndex")Integer startIndex,  
21.        @Param("pageSize")Integer pageSize  
22.    );  
23.  
24.    public Integer queryDrawingNumByAuthorId(  
25.        @Param("authorId")Long id,  
26.        @Param("title")String title,  
27.        @Param("label")String label,  
28.        @Param("classifyId")Integer classifyId,  
29.        @Param("state")Integer state  
30.    );  
31.  
32.    /**  
33.     * 查询指定 id 的手绘  
34.     * @param id  
35.     * @return  
36.     */  
37.    public DrawingEntity queryDrawingById(@Param("id")Long id);  
38.  
39.    /**  
40.     * 查询符合条件的手绘
```

```

41.      * @param title
42.      * @param label
43.      * @param classifyId
44.      * @param state
45.      * @param startIndex
46.      * @param pageSize
47.      * @param likeNum
48.      * @return
49.      */
50.      public List<DrawingEntity> queryDrawingByTitleLabClassifyState(
51.          @Param("title")String title,
52.          @Param("label")String label,
53.          @Param("classifyId")Integer classifyId,
54.          @Param("state")Integer state,
55.          @Param("startIndex")Integer startIndex,
56.          @Param("pageSize")Integer pageSize,
57.          @Param("likeNum")Integer likeNum
58.      );
59.
60.      /**
61.       * 返回上个方法的符合条件的个数
62.       * @param title
63.       * @param label
64.       * @param classifyId
65.       * @param state
66.       * @param likeNum
67.       * @return
68.       */
69.      public Integer queryDrawingNumByTitleLabClassifyState(
70.          @Param("title")String title,
71.          @Param("label")String label,
72.          @Param("classifyId")Integer classifyId,
73.          @Param("state")Integer state,
74.          @Param("likeNum")Integer likeNum
75.      );
76.
77.      /**
78.       * 更新指定手绘的状态
79.       * @param id
80.       * @param state
81.       * @return
82.       */
83.      public Integer updateDrawingState(@Param("id")Long id, @Param("state")Integer state);

```

```

84.
85.     /**
86.      * 更新点赞数
87.      * @param id
88.      * @param num 需要添加的点赞数
89.      * @return
90.      */
91.     public Integer updateLikeNumById(@Param("id")Long id, @Param("num")Integer num);
92. }

```

4.10.8 HomeMenuMapper

```

1. @Mapper
2. @Repository
3. public interface HomeMenuMapper {
4.
5.     /**
6.      * 插入一条新的 homeMenu
7.      * @param homeMenuEntity
8.      * @return
9.      */
10.    public Integer insertMenu(HomeMenuEntity homeMenuEntity);
11.
12.    /**
13.     * 查询全部 homeMenu
14.     * @return
15.     */
16.    public List<HomeMenuEntity> queryMenuByState(@Param("state")Integer state);
17.
18.    /**
19.     * 更新 menu 状态
20.     * @param id
21.     * @param state
22.     * @return
23.     */
24.    public Integer updateMenuState(@Param("id")Integer id, @Param("state")Integer state);
25.
26. }

```

4.10.9 LogMapper

```

1. @Mapper

```

```

2. @Repository
3. public interface LogMapper {
4.
5.     /**
6.      * 插入一条 log
7.      * @param logEntity
8.      * @return
9.      */
10.    public Integer insertLog(LogEntity logEntity);
11.
12.    /**
13.     * 查询指定页的 log
14.     * @param startIndex
15.     * @param pageSize
16.     * @return
17.     */
18.    public List<LogEntity> queryLog(
19.        @Param("startIndex")Integer startIndex,
20.        @Param("pageSize")Integer pageSize
21.    );
22.
23.    public Integer queryLogNum();
24.
25. }

```

4.10.10 RoleMapper

```

1. @Mapper
2. @Repository
3. public interface RoleMapper {
4.
5.     /**
6.      * 插入新角色
7.      * @param roleEntity
8.      * @return
9.      */
10.    public Integer insertRole(RoleEntity roleEntity);
11.
12.    /**
13.     * 查询全部角色
14.     * @param state
15.     * @return
16.     */
17.    public List<RoleEntity> queryRole(@Param("state")Integer state);
18.

```

```

19.  /**
20.   * 查询指定 ids 的角色
21.   * @author mxf
22.   * @param ids
23.   * @param state
24.   * @return
25.   */
26.  public List<RoleEntity> queryRoleByIds(@Param("ids")List<Integer> ids, @
    Param("state")Integer state);
27.
28.  /**
29.   * 更改角色状态
30.   * @param id
31.   * @param state
32.   * @return
33.   */
34.  public Integer updateRoleState(@Param("id")Integer id, @Param("state")In
    teger state);
35.
36. }

```

4.10.11 RoleMenuMapper

```

1. @Mapper
2. @Repository
3. public interface RoleMenuMapper {
4.
5.  /**
6.   * 插入新角色及其菜单
7.   * @param roleId
8.   * @param menuId
9.   * @return
10.  */
11.  public Integer insertRoleMenu(
12.      @Param("roleId")Integer roleId,
13.      @Param("menuIds")Integer[] menuId
14.  );
15.
16.  public Integer deleteRoleMenuByRoleMenuId(@Param("roleMenuId")Integer ro
    leMenuId);
17.
18.  /**
19.   * 查询指定角色对应的菜单
20.   * @param roleIds
21.   * @return

```

```

22.     */
23.     public List<RoleMenuEntity> queryByRoleIds(@Param("roleIds")List<Integer
    > roleIds);
24.
25.     /**
26.      * 查询指定角色对应的菜单 id
27.      * @param ids
28.      * @return
29.      */
30.     public List<Integer> queryMenuIdByRoleIds(@Param("roleIds")List<Integer>
    roleIds);
31.
32. }

```

4.10.12 ThemeMapper

```

1. @Mapper
2. @Repository
3. public interface ThemeMapper {
4.
5.     /**
6.      * 插入新主题
7.      * @param themeEntity
8.      * @return
9.      */
10.    public Integer insertNewTheme(ThemeEntity themeEntity);
11.
12.    /**
13.     * 删除指定主题
14.     * @param id
15.     * @return
16.     */
17.    public Integer deleteThemeById(@Param("id")Integer id);
18.
19.    /**
20.     * 获取全部主题
21.     * @return
22.     */
23.    public List<ThemeEntity> queryTheme();
24.
25.    /**
26.     * 获取指定主题
27.     * @param id
28.     * @return
29.     */

```



```

30.     public ThemeEntity queryThemeById(@Param("id")Integer id);
31.
32.     /**
33.      * 设定默认主题，原默认主题会被设为非默认
34.      * @param id
35.      * @return
36.      */
37.     public Integer setDefaultTheme(@Param("id")Integer id);
38.
39. }

```

4.10.13 UserMapper

```

1. @Mapper
2. @Repository
3. public interface UserMapper {
4.
5.     /**
6.      * 插入新用户信息
7.      * @param userEntity
8.      */
9.     public Integer insertUser(UserEntity userEntity);
10.
11.    /**
12.     * 返回指定 id 的用户信息
13.     * @param id
14.     * @return
15.     */
16.    public UserEntity queryUserById(@Param("id")Long id);
17.
18.    /**
19.     * 查询符合条件的 user
20.     * @param telephone
21.     * @param nickname
22.     * @param state
23.     * @param vip
24.     * @param startIndex
25.     * @param pageSize
26.     * @return
27.     */
28.    public List<UserEntity> queryUser(
29.        @Param("telephone")String telephone,
30.        @Param("nickname")String nickname,
31.        @Param("state")Integer state,
32.        @Param("vip")Integer vip,

```

```
33.         @Param("startIndex")Integer startIndex,
34.         @Param("pageSize")Integer pageSize
35.     );
36.
37.     /**
38.      * 返回上一方法的总数
39.      * @param telephone
40.      * @param nickname
41.      * @param state
42.      * @param vip
43.      * @return
44.      */
45.     public Integer queryUserNum(
46.         @Param("telephone")String telephone,
47.         @Param("nickname")String nickname,
48.         @Param("state")Integer state,
49.         @Param("vip")Integer vip
50.     );
51.
52.     /**
53.      * 通过手机号查询用户
54.      * @param telephone
55.      * @return
56.      */
57.     public UserEntity queryUserByTel(@Param("telephone")String telephone);
58.
59.     /**
60.      * 修改用户信息
61.      * @param userEntity
62.      */
63.     public Integer updateUser(UserEntity userEntity);
64.
65.     /**
66.      * 更新用户状态
67.      * @param id
68.      * @param state 0 正常 1 冻结
69.      */
70.     public Integer updateUserState(@Param("id")Long id, @Param("state")int s
    tate);
71.
72.     /**
73.      * 更新用户 vip 状态
74.      * @param id
75.      * @param vip 0 普通 1vip
```

```
76.     */
77.     public Integer updateUserVip(@Param("id")Long id, @Param("vip")int vip)
78. }
```

4.11 bo 包

4.11.1 AdminBo

```
1. public class AdminBO {
2.     private String account;
3.     private Integer id;
4.     private String password;
5.     private String roleName;
6. }
7.
```

4.11.2 BackMenuBo

```
1. public class BackMenuBO {
2.
3.     private Integer id;
4.     private String name;
5.     private String path;
6.     private BackMenuBO[] sonMenu;
7.
8.     @JsonIgnore
9.     private List<BackMenuBO> sonMenuList;
10.
11.     public void addSonMenuList(BackMenuBO backMenuBO);
12.     public void tranArray();
13.
14. }
```

4.11.3 BusinessObject

```
1. public interface BusinessObject {
2.
3. }
```

4.11.4ClassifyAdminBo

```
1. public class ClassifyAdminBO {
2.     private Integer id;
3.     private String name;
```

```
4.     private Integer status;
5. }
```

4.11.5 ClassifyBo

```
1. public class ClassifyBO {
2.
3.     private Integer id;
4.     private String name;
5. }
```

4.11.6 DetailAuthorBo

```
1. public class DetailAuthorBO implements BusinessObject {
2.
3.     private Long id;
4.     private String nickname;
5.     private String avatar;
6.
7. }
```

4.11.7 HomeMenuBo

```
1. public class HomeMenuBO implements BusinessObject {
2.
3.     private Integer id;
4.     private String name;
5.     private String path;
6.     private Integer state;
7.
8.     public static List<HomeMenuBO> transMenuEntityToVo(List<HomeMenuEntity>
    homeMenuEntities);
9. }
```

4.11.8 LogBo

```
1. public class LogBo {
2.     private String dateTime;
3.     private String content;
4.     private String adminId;
5.
6.     public static List<LogBo> transLogEntityToBo(List<LogEntity> logs);
7. }
```

4.11.9 MyArticleBo

```

1. public class MyArticleBO {
2.
3.     private Long id;
4.     private String cover;
5.     private String title;
6.     private String introduction;
7.     private String dateTime;
8.     private Integer likeNum;
9.     private Integer comment;
10.    private Integer status;
11. }

```

4.11.10 MyDrawingBo

```

1. public class MyDrawingBO {
2.
3.     private Long id;
4.     private String cover;
5.     private String title;
6.     private String introduction;
7.     private String dateTime;
8.     private Integer likeNum;
9.     private Integer comment;
10.    private Integer status;
11.
12. }

```

4.11.11 Result

```

1. public class Result {
2.     private Integer code = 1;
3.     private String message = null;
4.
5.     /**
6.      * 默认正常情况 code = 1 message = null
7.      */
8.     public Result() {
9.     }
10.    /**
11.     * 出现错误 code =0 message = 错误信息
12.     */
13.    public Result(String err)
14. }

```

4.11.12 SearchArticleBo

```
1. public class SearchArticalBO implements BusinessObject {
2.
3.     private Long id;
4.     private String cover;
5.     private String title;
6.     private String introduction;
7.     private String avatar;
8.     private String nickname;
9.     private String dateTime;
10.    private Integer likeNum;
11.    private Integer comment;
12.    private Integer status;
13.    private String classifyName;
14. }
```

4.11.13 SearchDrawingBo

```
1. public class SearchDrawingBO implements BusinessObject {
2.
3.     private Long id;
4.     private String cover;
5.     private String title;
6.     private String introduction;
7.     private String avatar;
8.     private String nickname;
9.     private String dateTime;
10.    private Integer likeNum;
11.    private Integer comment;
12.    private Integer status;
13.    private String classifyName;
14. }
15.
```

4.11.14 ThemeListBo

```
1. public class ThemeListBO {
2.
3.     private Integer id;
4.     private String name;
5.     private String bg;
6.     private String bgm;
7.     private Integer isdefault;
8. }
```

4.11.15 UserBo

```

1. public class UserBO {
2.
3.     private String id;
4.     private String nickName;
5.     private String telephone;
6.     private Integer sex;
7.     private String email;
8.     private Integer status;
9.     private Integer vip;
10.    private String createTime;
11.    public static List<UserBO> transToVo(List<UserEntity> userEntities);
12. }

```

4.12 vo 包

4.12.1 request

4.12.1.1 AddClassifyAdminRequestVo

```

1. public class AddClassifyAdminRequestVO {
2.
3.     private String name;
4.     private Integer type;
5.
6. }

```

4.12.1.2 AdminLoginRequestVo

```

1. public class AdminLogInRequestVO implements ViewObject {
2.
3.     private String account;
4.     private String password;
5.
6. }

```

4.12.1.3 AdminRegisterRequestVo

```

1. public class AdminRegisterRequestVO implements ViewObject {
2.
3.     private String account;
4.     private String password;
5.     private Integer roleId;
6.
7. }

```

4.12.1.4 ArticleVo

```
1.  public class ArticleVO extends WorkVO {
2.      private String cover;
3.      private String title;
4.      private Integer classify;
5.      private String[] labels;
6.      private String introduction;
7.      private String fileUrl;
8.
9.  }
```

4.12.1.5 DeleteWorkRequest

```
1.  public class DeleteWorkRequestVO {
2.
3.      private Long id;
4.      private Integer type;
5.  }
```

4.12.1.6 DrawingVo

```
1.  public class DrawingVO extends WorkVO{
2.
3.      private String title;
4.      private Integer classify;
5.      private String[] labels;
6.      private String introduction;
7.      private String[] image;
8.  }
```

4.12.1.7 GetVerificationCodeRequestVo

```
1.  public class GetVerificationCodeRequestVO implements ViewObject {
2.
3.      private String telephone;
4.
5.  }
6.
```

4.12.1.8 RegisterRequestVo

```
1.  public class RegisterRequestVO implements ViewObject {
2.
3.      private String nickname;
```



```
4.     private String pwd;
5.     private String telephone;
6.     private String code;
7.
8. }
```

4.12.1.9 ThemeVo

```
1.  public class ThemeVO {
2.      private String name;
3.      private String bg;
4.      private String bgm;
5.
6. }
```

4.12.1.10 UserLoginRequestVo

```
1.  public class UserLogInRequestVO implements ViewObject{
2.
3.      private String telephone;
4.      private String pwd;
5.
6. }
```

4.12.1.11 UserVo

```
1.  public class UserVo {
2.      private String nickName;
3.      private Integer sex;
4.      private String email;
5.      private String sign;
6.      private String avatar;
7.      private Long id;
8. }
```

4.12.1.12 WorkVo

```
1.  public abstract class WorkVO {
2. }
```

4.12.2 response

4.12.2.1 AdminLogInVo

```
1.  public class AdminLogInVO implements ViewObject {
2. 
```

```
3.     private String token;
4.     private BackMenuBO[] menu;
5.
6. }
```

4.12.2.2 ArticleVo

```
1.  public class ArticleVo {
2.      private Long id;
3.      private String title;
4.      private String cover;
5.      private String introduction;
6.      private String avatar;
7.      private String nickName;
8.      private Timestamp dateTime;
9.      private String comment;
10.     private Integer likeNum;
11.     private Integer status;
12.     private String classifyName;
13. }
```

4.12.2.3 DrawingVo

```
1.  public class DrawingVo {
2.      private Long id;
3.      private String title;
4.      private String cover;
5.      private String introduction;
6.      private String avatar;
7.      private String nickName;
8.      private Timestamp dateTime;
9.      private String comment;
10.     private Integer likeNum;
11.     private Integer status;
12.     private String classifyName;
13. }
```

4.12.2.4 GetClassifyAdminVo

```
1.  public class GetClassifyAdminVO implements ViewObject {
2.
3.      private ClassifyAdminBO[] classifys;
4.      private Integer totalPage;
5. }
```

4.12.2.5 GetDetailArticleVo

```
1.  public class GetDetailArticleVO implements ViewObject {
2.
3.      private DetailAuthorBO author;
4.      private String title;
5.      private String cover;
6.      private String introduction;
7.      private String fileUrl;
8.      private Integer likeNum;
9.      private Integer isLike;
10.     private Integer isCollect;
11.     private String[] labels;
12.     private Integer type = 0;
13.     private String dateTime;
14.     private Integer status;
15.     private String classifyName;
16.
17. }
18.
```

4.12.2.6 MyWorksArticleVo

```
1.  public class MyWorksArticleVO implements ViewObject {
2.
3.      private MyArticleBO[] articles;
4.      private Integer totalPage;
5.
6. }
```

4.12.2.7 MyWorksDrawingVo

```
1.  public class MyWorksDrawingVO implements ViewObject {
2.
3.      private MyDrawingBO[] drawing;
4.      private Integer totalPage;
5.
6. }
```

4.12.2.8 ResponseVo

```
1.  public class ResponseVO implements ViewObject {
2.
3.      private ViewObject data;
4.      private Result result;
```

```
5.  
6. }
```

4.12.2.9 SearchWorkArticleVo

```
1.  public class SearchWorkArticleVO implements ViewObject {  
2.  
3.      private SearchArticalBO[] articles;  
4.      private Integer totalPage;  
5.  
6. }
```

4.12.2.10 SearchWorkDrawingVo

```
1.  public class SearchWorkDrawingVO implements ViewObject {  
2.  
3.      private SearchDrawingBO[] drawing;  
4.      private Integer totalPage;  
5.  
6. }
```

4.12.2.11 ThemeListVo

```
1.  public class ThemeListVO implements ViewObject {  
2.  
3.      private ThemeListBO[] themes;  
4.  
5. }
```

4.12.2.12 UploadFileVo

```
1.  public class UploadFileVO implements ViewObject {  
2.  
3.      private String url;  
4.  
5. }
```

4.12.2.13 UserLoginVo

```
1.  public class UserLogInVO implements ViewObject {  
2.  
3.      private String token;  
4.      private Long id;  
5.      private HomeMenuBO[] menus;  
6.
```

```
7. }
```

4.12.2.14 GetDetailDrawingVo

```
1. public class GetDetailDrawingVO implements ViewObject {
2.
3.     private DetailAuthorBO author;
4.     private String title;
5.     private String introduction;
6.     private String[] images;
7.     private Integer likeNum;
8.     private Integer isLike;
9.     private Integer isCollect;
10.    private String[] labels;
11.    private Integer type = 1;
12.    private String dateTime;
13.    private Integer status;
14.    private String classifyName;
15.
16. }
```

4.12.3 ViewObject

```
1. public interface ViewObject {
2.
3. }
```

4.13 entity 包

4.13.1 AdminEntity

```
1. public class AdminEntity {
2.
3.     private Integer id;
4.     private String account;
5.     private String password;
6.     private Integer state;
7.     private Timestamp createTime;
8.     private Timestamp updateTime;
9.
10. }
```

4.13.2 AdminRoleEntity

```
1. public class AdminRoleEntity {  
2.  
3.     private Integer adminRoleId;  
4.     private Integer adminId;  
5.     private Integer roleId;  
6.     private Timestamp createTime;  
7.  
8. }
```

4.13.3 ArticleEntity

```
1. public class ArticleEntity extends Work {  
2.  
3.     private Long id;  
4.     private Long authorId;  
5.     private String title;  
6.     private String cover;  
7.     private String introduction;  
8.     private String url;  
9.     private String labels;  
10.    private Integer classifyId;  
11.    private Integer likeNum;  
12.    private Integer state;  
13.    private Timestamp createTime;  
14.    private Timestamp updateTime;  
15. }
```

4.13.4 BackMenuEntity

```
1. public class BackMenuEntity {  
2.  
3.     private Integer id;  
4.     private String name;  
5.     private String path;  
6.     private Integer parentId;  
7.     private Timestamp createTime;  
8.  
9. }
```

4.13.5 ClassifyArticleEntity

```
1. public class ClassifyArticleEntity extends ClassifyEntity {  
2.
```

```
3. }
```

4.13.6 ClassifyDrawingEntity

```
1. public class ClassifyDrawingEntity extends ClassifyEntity {  
2.  
3. }
```

4.13.7 ClassifyEntity

```
1. public class ClassifyEntity {  
2.  
3.     private Integer id;  
4.     private String name;  
5.     private Integer state;  
6.     private Timestamp createTime;  
7.     private Timestamp updateTime;  
8.  
9. }
```

4.13.8 DrawingEntity

```
1. public class DrawingEntity extends Work {  
2.  
3.     private Long id;  
4.     private Long authorId;  
5.     private String title;  
6.     private String introduction;  
7.     private String labels;  
8.     private String url;  
9.     private Integer classifyId;  
10.    private Integer state;  
11.    private Integer likeNum;  
12.    private Timestamp createTime;  
13.    private Timestamp updateTime;  
14.  
15. }
```

4.13.9 HomeMenuEntity

```
1. public class HomeMenuEntity {  
2.  
3.     private Integer id;  
4.     private String name;  
5.     private String path;
```

```
6.     private Integer parentId;
7.     private Integer state;
8.     private Timestamp createTime;
9.     private Timestamp updateTime;
10. }
```

4.13.10 LogEntity

```
1. public class LogEntity {
2.
3.     private Long id;
4.     private Integer adminId;
5.     private String content;
6.     private Timestamp createTime;
7.
8. }
```

4.13.11 RoleMenuEntity

```
1. public class RoleMenuEntity {
2.
3.     private Integer roleMenuId;
4.     private Integer roleId;
5.     private Integer menuId;
6.     private Timestamp createTime;
7. }
```

4.13.12 RoleEntity

```
1. public class RoleEntity {
2.
3.     private Integer id;
4.     private String name;
5.     private String introduction;
6.     private Integer state;
7.     private Timestamp createTime;
8.     private Timestamp updateTime;
9. }
```

4.13.13 ThemeEntity

```
1. public class ThemeEntity {
2.
3.     private Integer id;
4.     private String name;
```



```
5.     private String backgroundUrl;
6.     private String bgmUrl;
7.     private Integer isDefault;
8.     private Timestamp createTime;
9.
10. }
```

4.13.14 UserEntity

```
1. public class UserEntity {
2.
3.     private Long id;
4.     private String telephone;
5.     private String password;
6.     private String nickname;
7.     private Integer gender;
8.     private String eMail;
9.     private String sign;
10.    private String avatar;
11.    private Integer vip;
12.    private Integer state;
13.    private Timestamp createTime;
14.    private Timestamp updateTime;
15. }
```

4.13.15 Work

```
1. public class Work {
2.
3. }
```

4.14 global 包

4.14.1 Global

```
1. public class Global {
2.
3.     final public static String letterTable = "123456789";
4.
5.     final public static Integer pageSize = 10;
6.
7. }
```

4.14.2 VerificationCode

```

1. public class VerificationCode {
2.
3.     public String code;
4.     private long genTime;
5.     private final long expireTime = 5 * 60;
6.
7.     public VerificationCode(String code){
8.         this.code = code;
9.         this.genTime = (new Date().getTime()) / 1000;
10.    }
11.
12.    public Boolean isExpire();
13.
14. }

```

4.14.3 VerificationCodeMap

```

1. public class VerificationCodeMap {
2.
3.     private Map<String, VerificationCode> map = new TreeMap<String, Verifica
        tionCode>();
4.     private static VerificationCodeMap instance = new VerificationCodeMap();
5.
6.     private VerificationCodeMap (){}
7.     public static VerificationCodeMap getInstance()
8.     public void put(String telephone, String code)
9.     public VerificationCode get(String telephone)
10.    public void remove(String telephone)
11. }

```

五、UI 设计

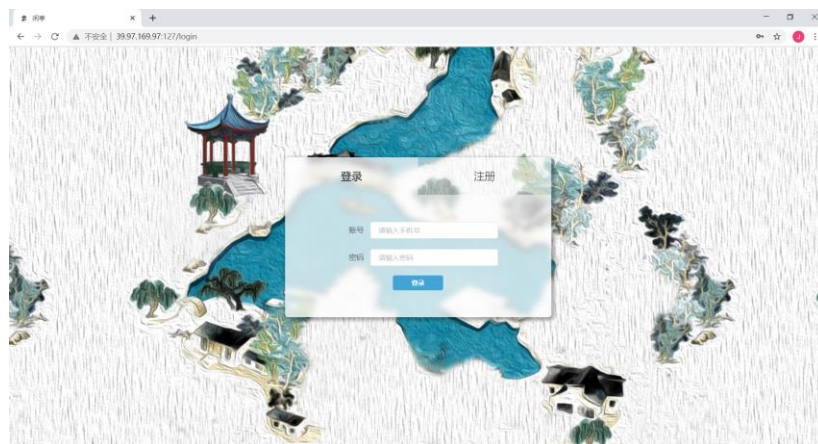
5.1 用户前台 UI

5.1.1 起始页:

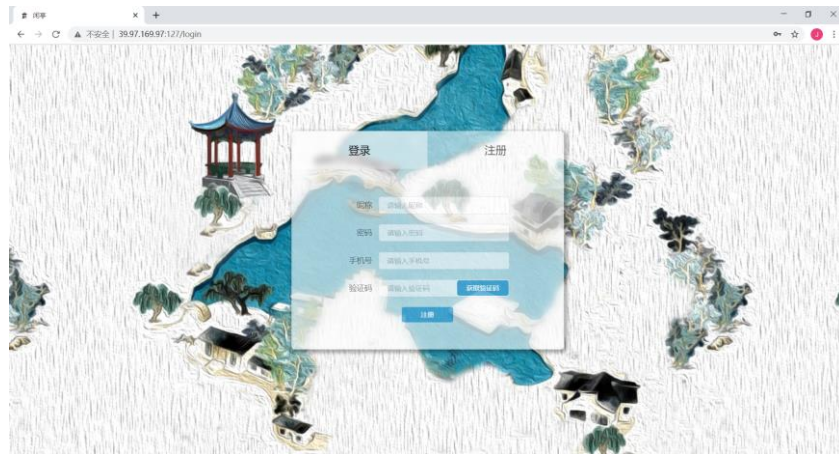


5.1.2 登录/注册:

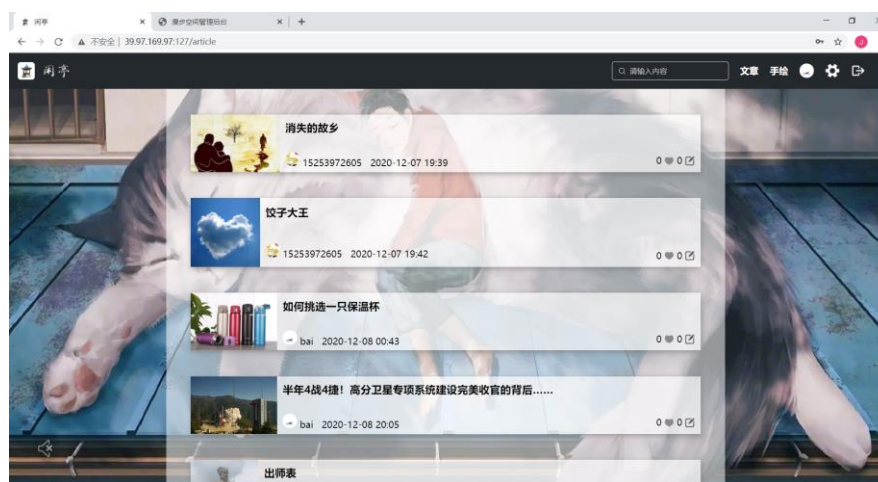
➤ 登录:



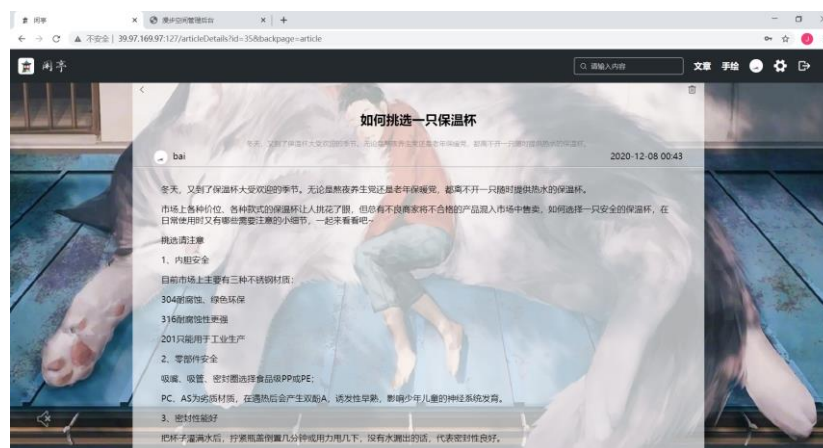
➤ 注册:



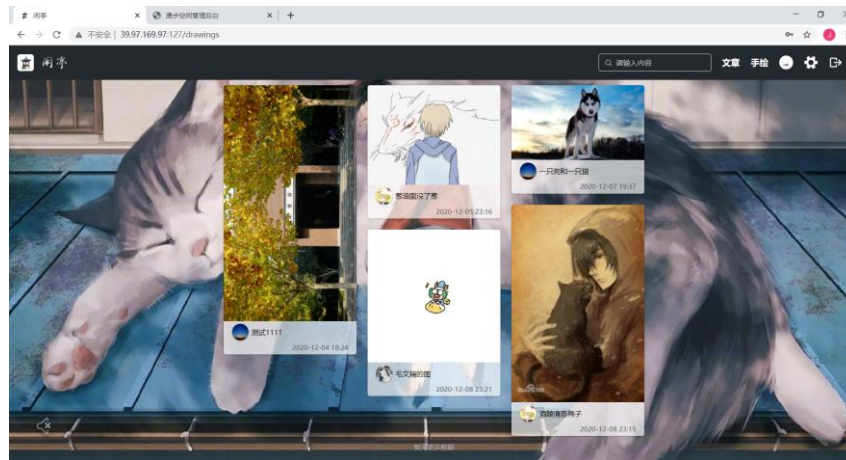
5.1.3 文章列表界面：



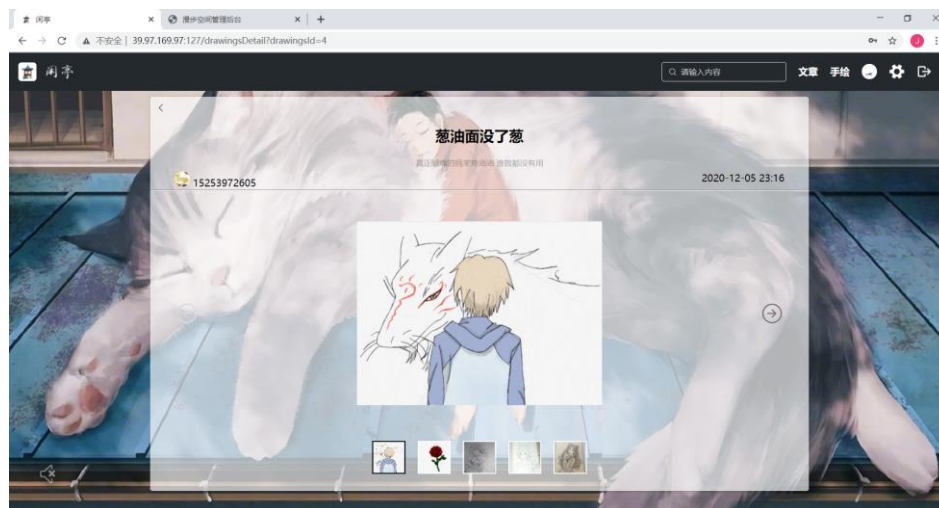
5.1.4 文章详情界面：



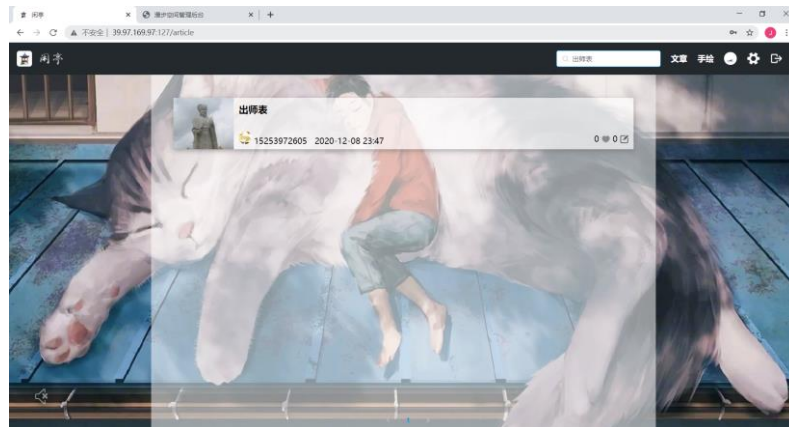
5.1.5 手绘列表界面：



5.1.6 手绘详情界面:



5.1.7 搜索：

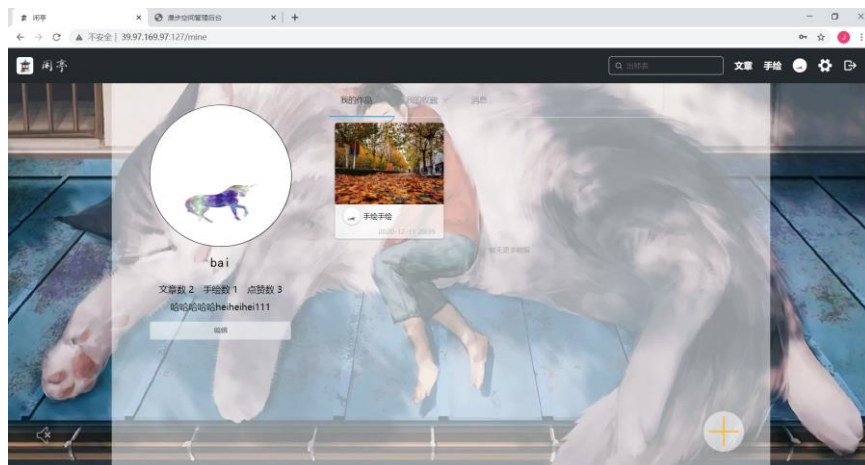


5.1.8 个人主页：

➤ 文章：



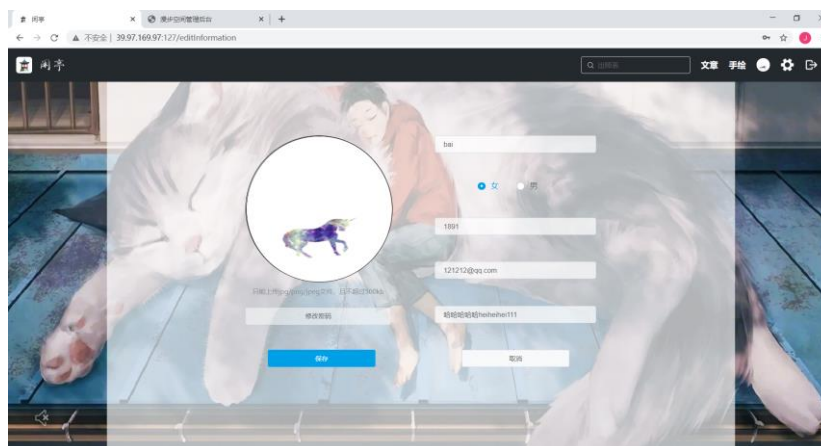
➤ 手绘：



➤ 消息：

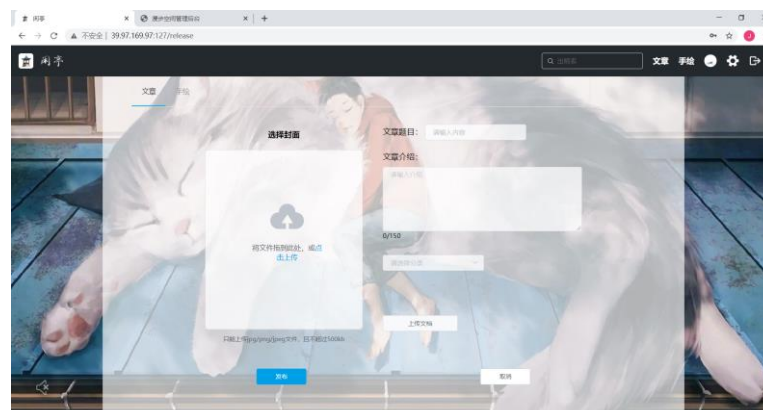


5.1.9 信息修改界面：

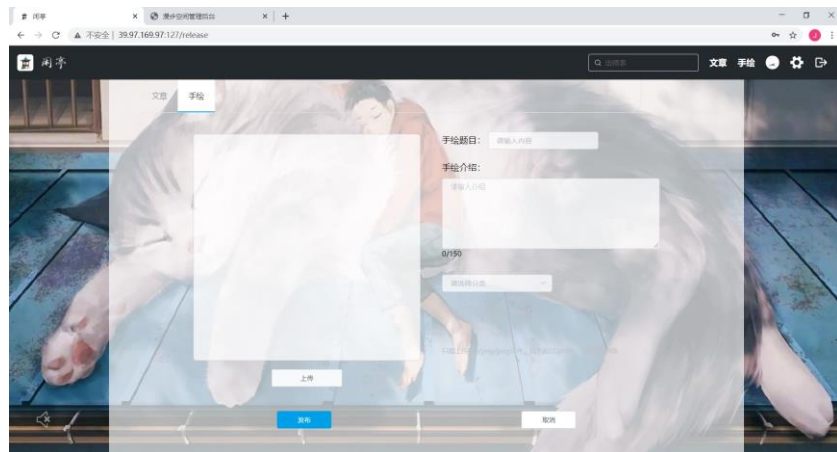


5.1.10 发布界面：

➤ 文章发布：

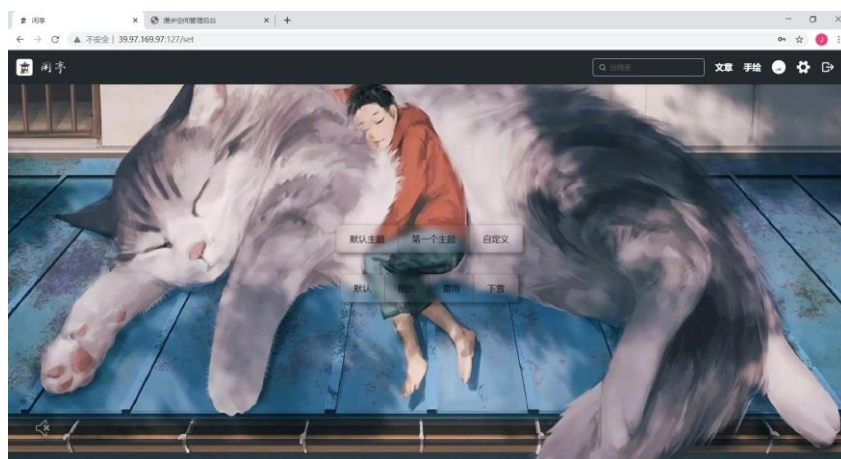


➤ 手绘发布：

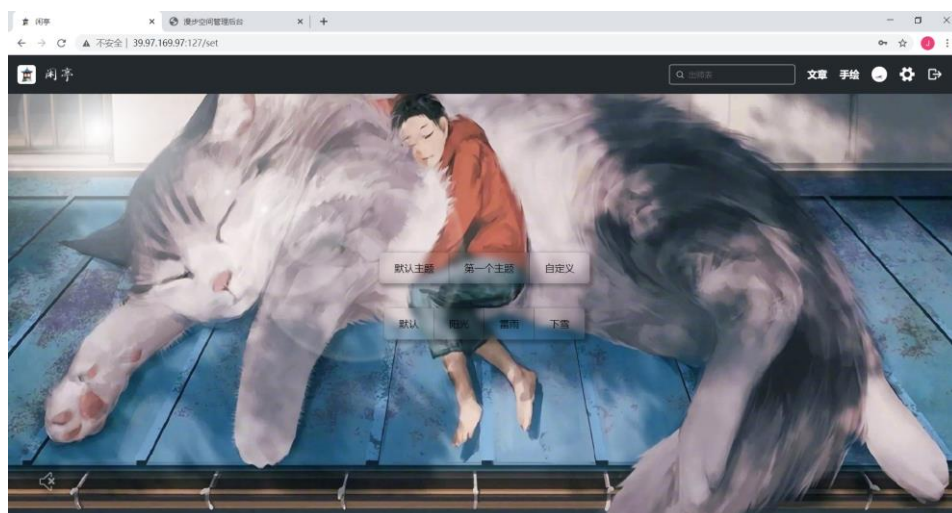


5.1.11 主题设置界面：

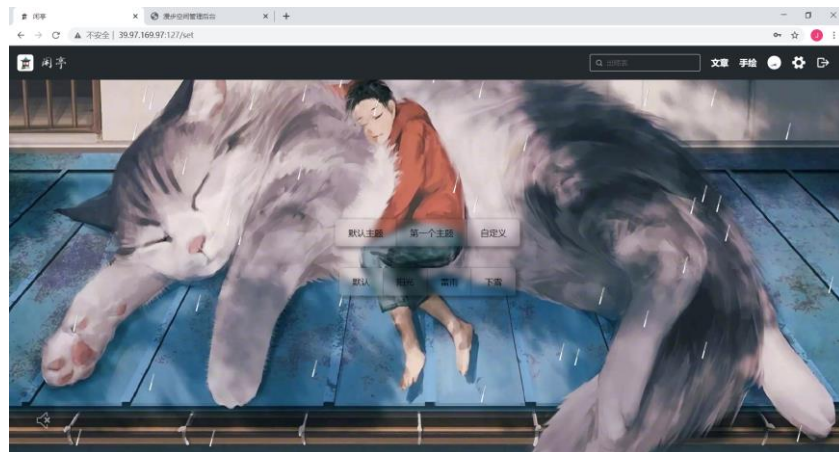
➤ 默认：



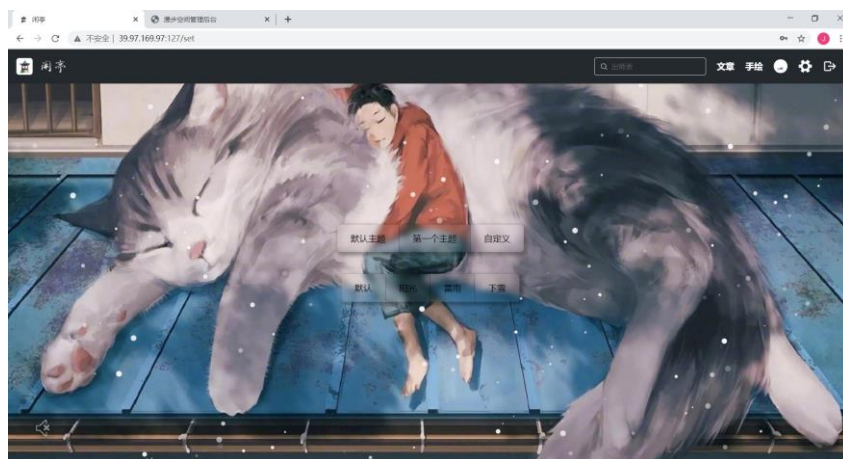
➤ 阳光：



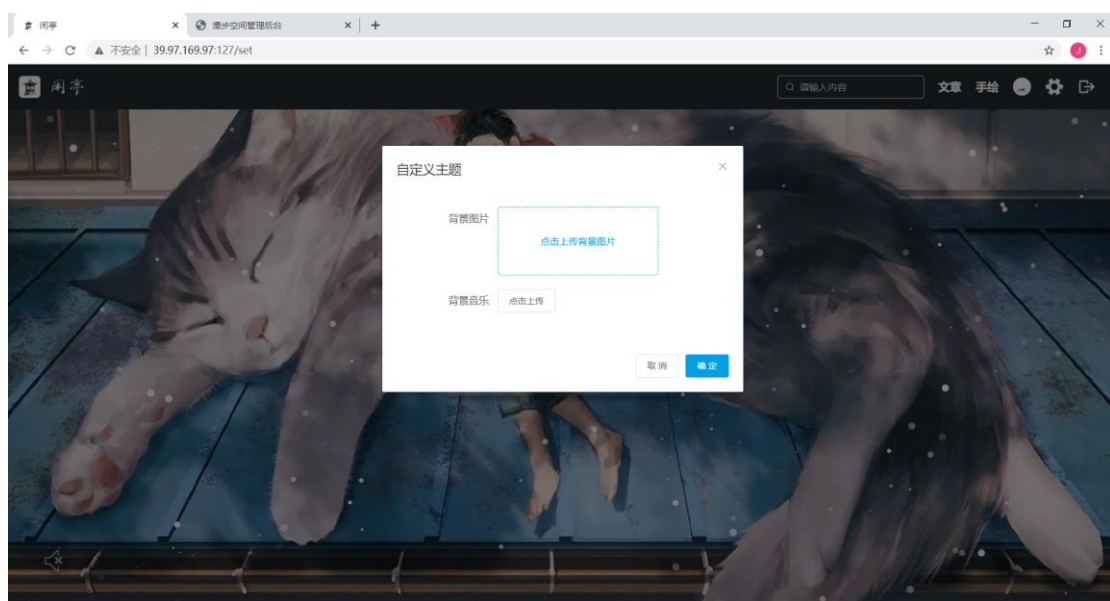
➤ 雷雨：



➤ 下雪：

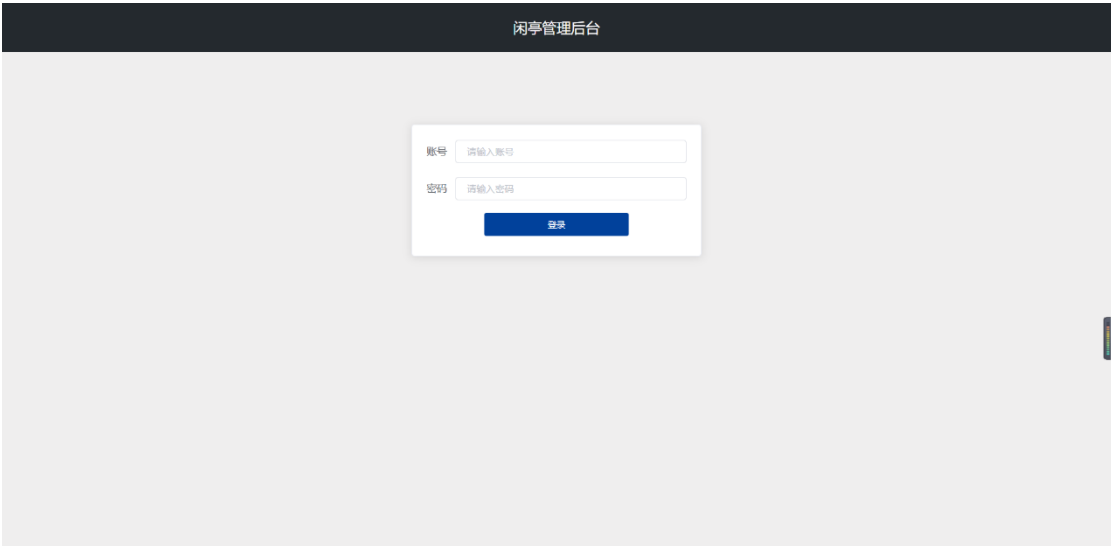


➤ 主题自定义：

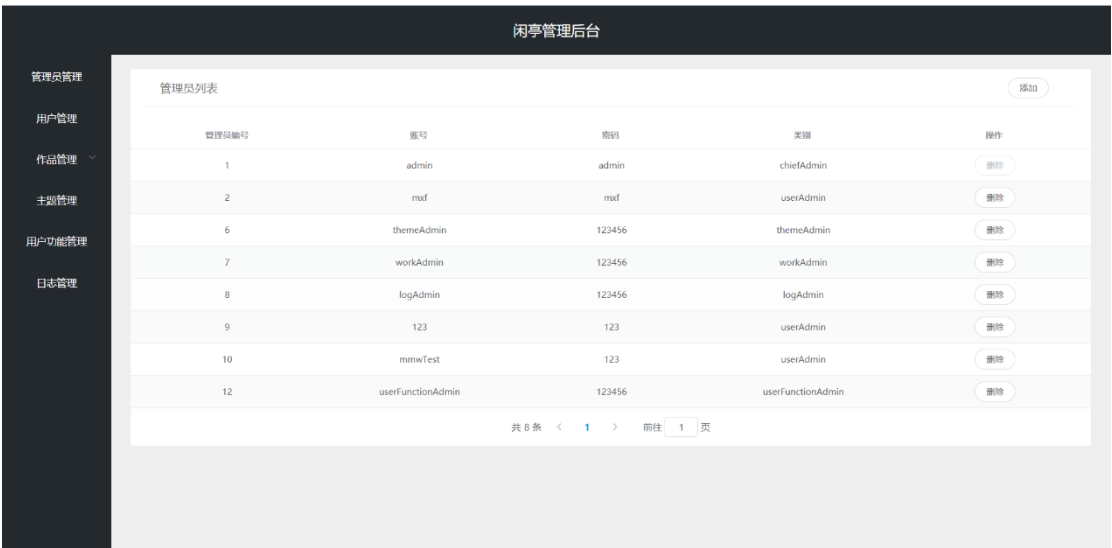


5.2 管理员后台 UI

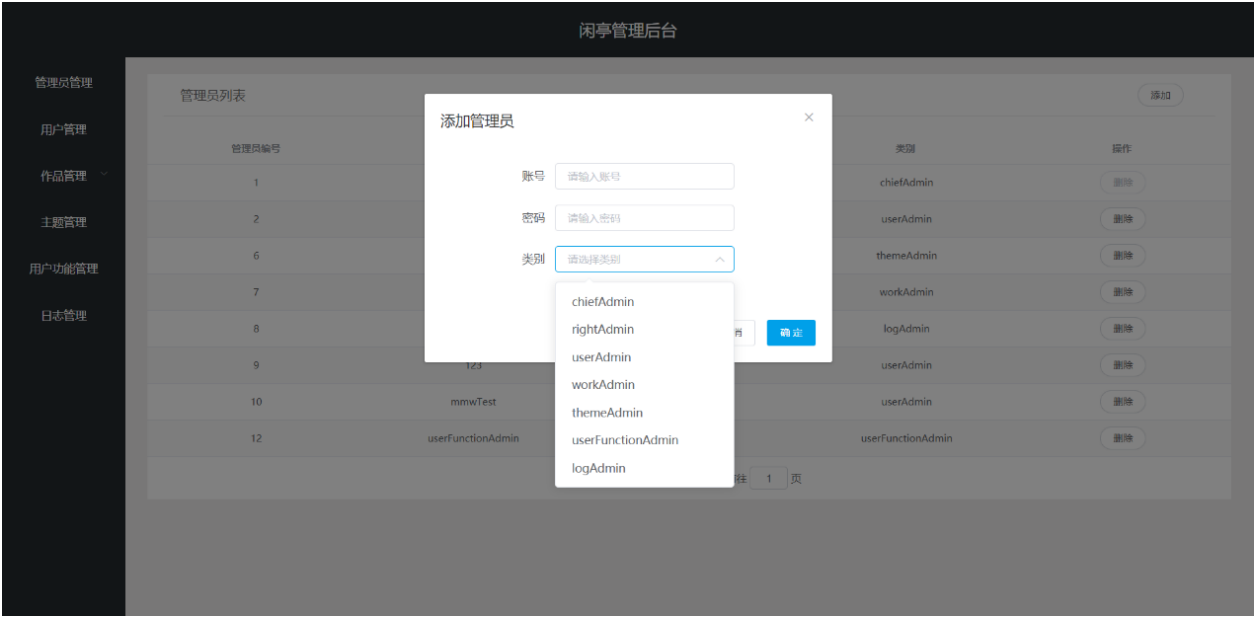
5.2.1 管理员登录界面



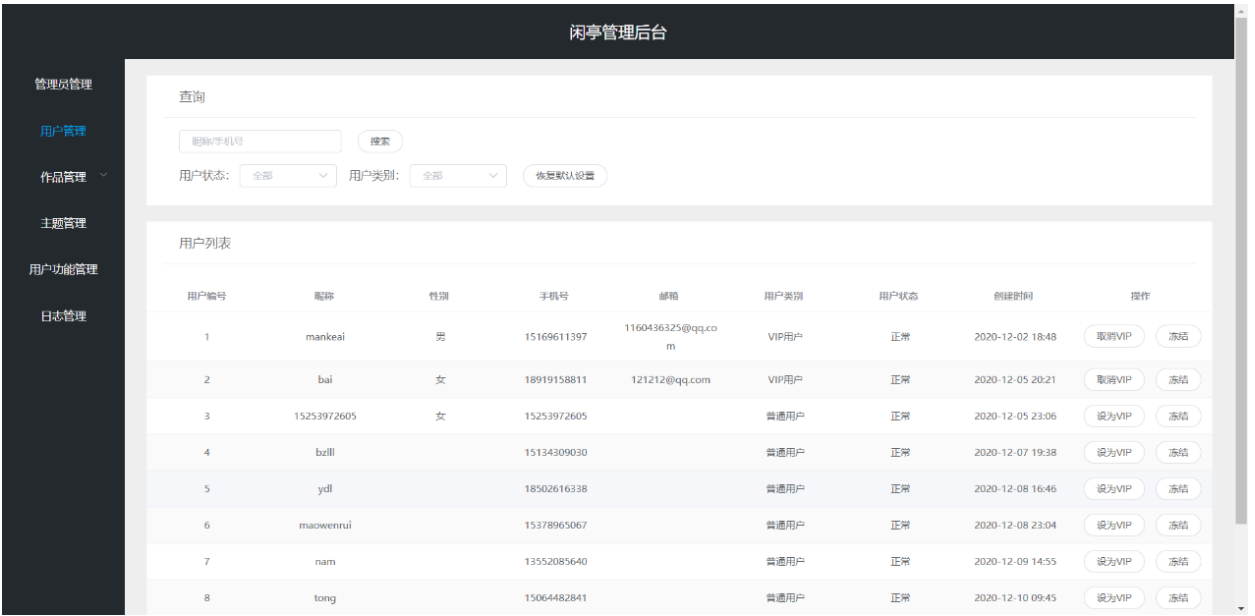
5.2.2 管理员管理



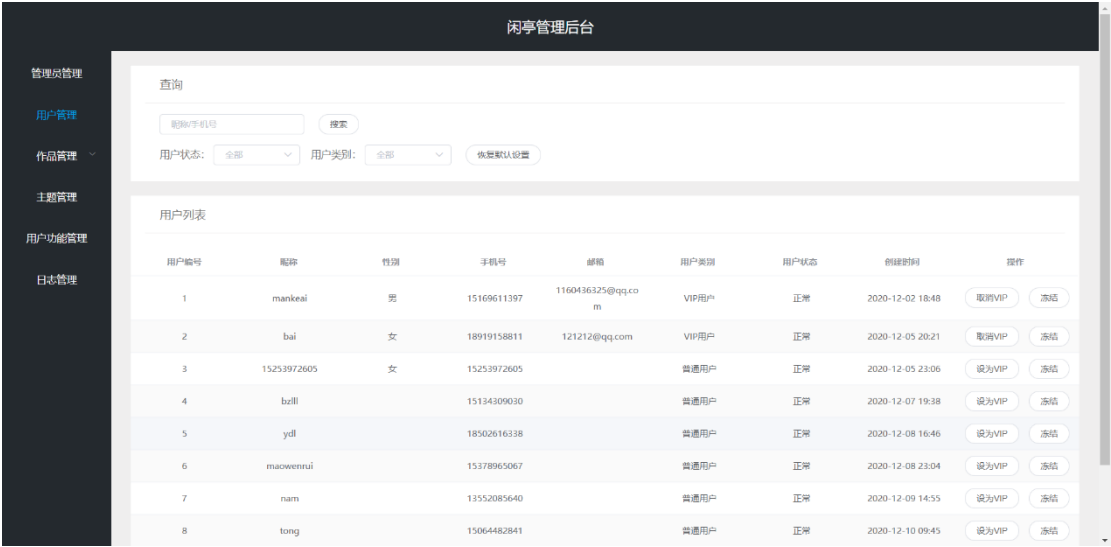
5.2.3 添加管理员



5.2.4 用户管理



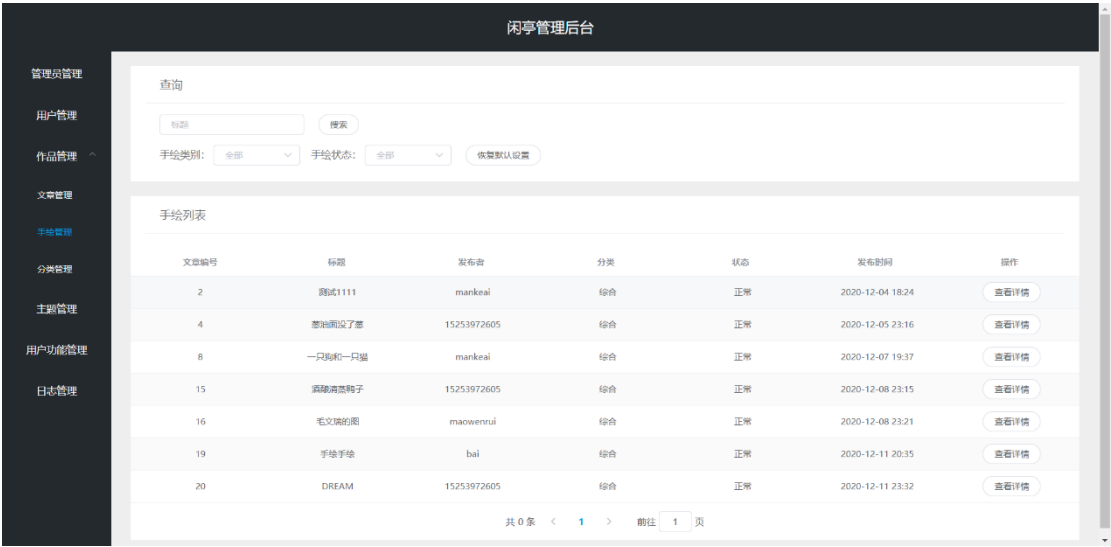
5.2.5 文章管理



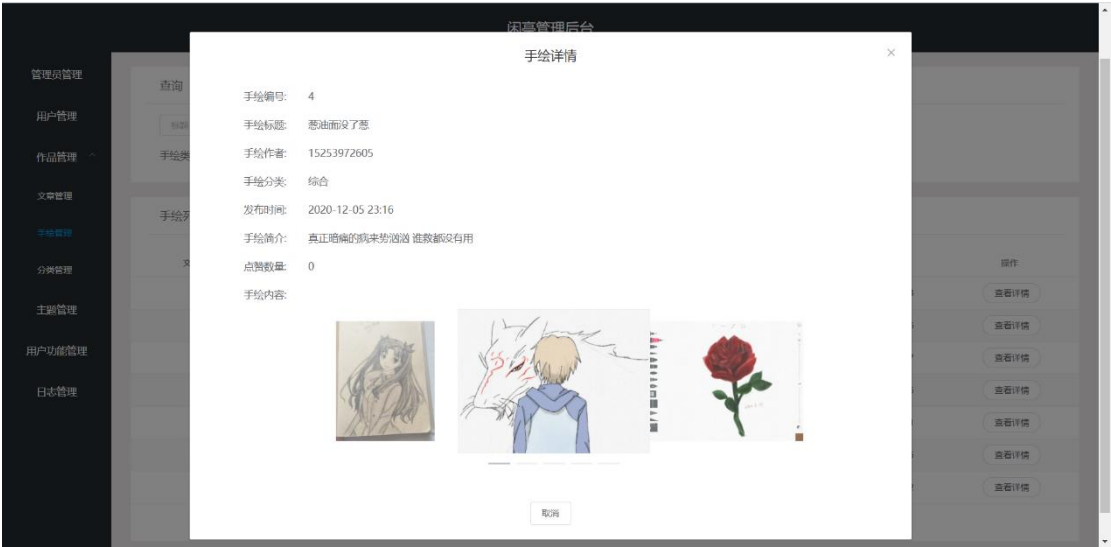
5.2.6 查看文章详情



5.2.7 手绘管理



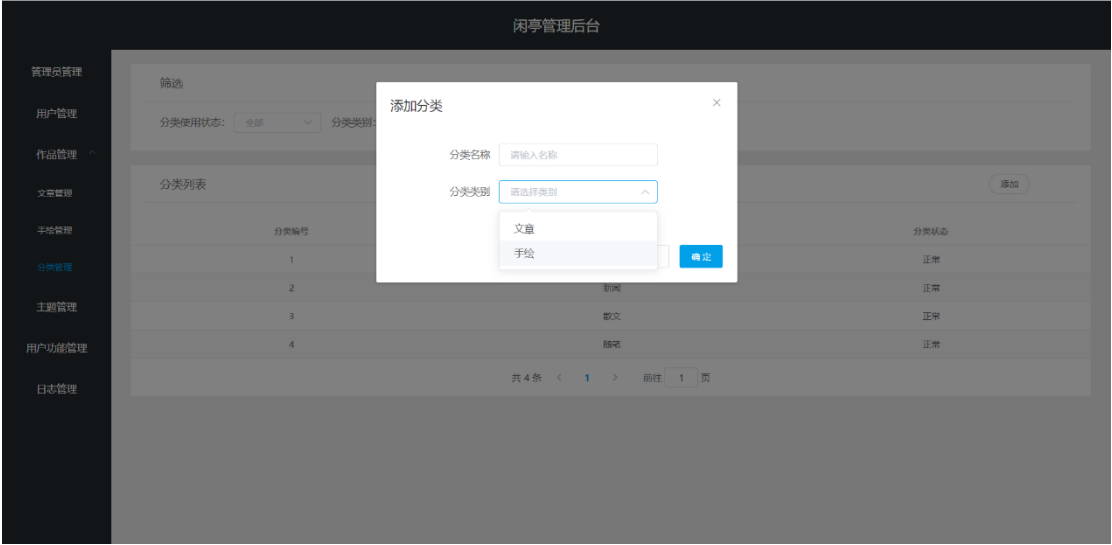
5.2.8 查看手绘详情



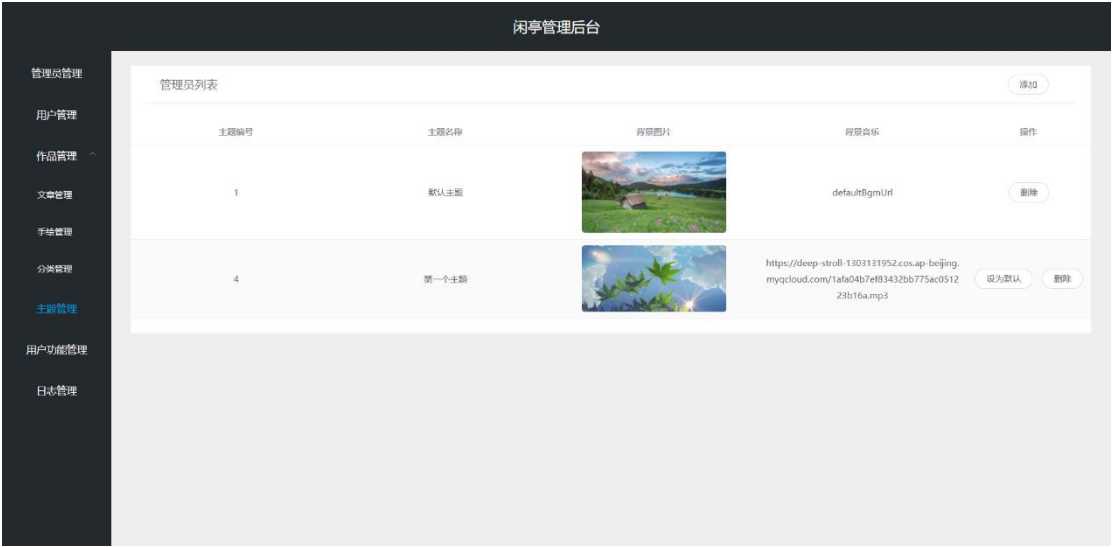
5.2.9 分类管理



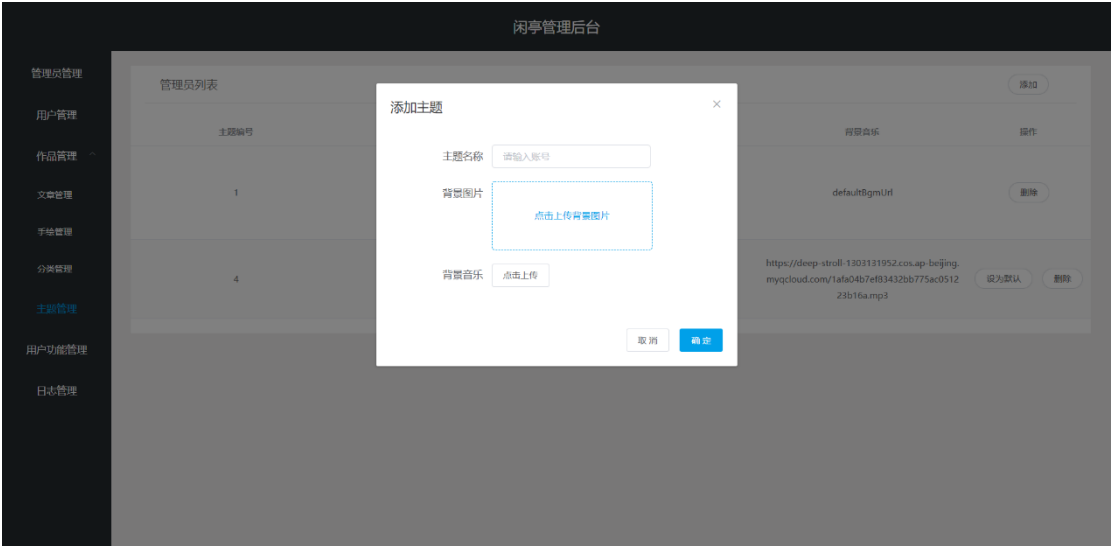
5.2.10 添加分类



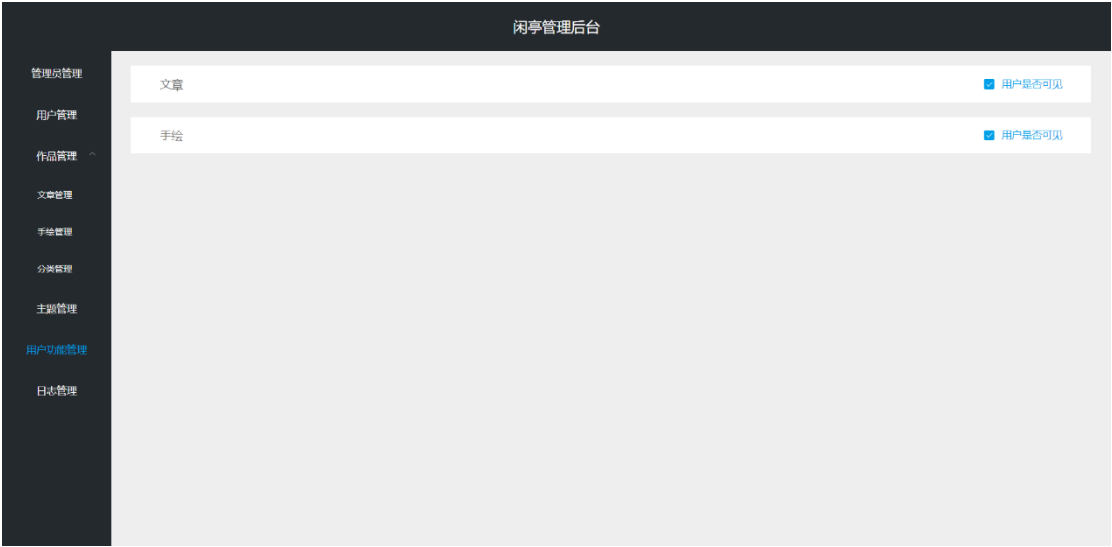
5.2.11 主题管理



5.2.12 添加主题



5.2.13 功能管理



5.2.14 日志管理

