

# Synapse平台系统建模报告

## 1. 引言

本次项目的建模报告覆盖了系统的核心功能，包括用户管理、内容创作、互动交流、系统管理等模块。绘制了多种UML图以展示项目的系统建模，包括领域模型、类图、用例图、序列图、流程图。由于上述图表**细节丰富，直接插入报告会损失细节**，不便于观察，因此所有**相关的UML图表已作为独立的图片打包在外部**，本报告将对这些图表进行**文字说明**和**设计原理的解释**。

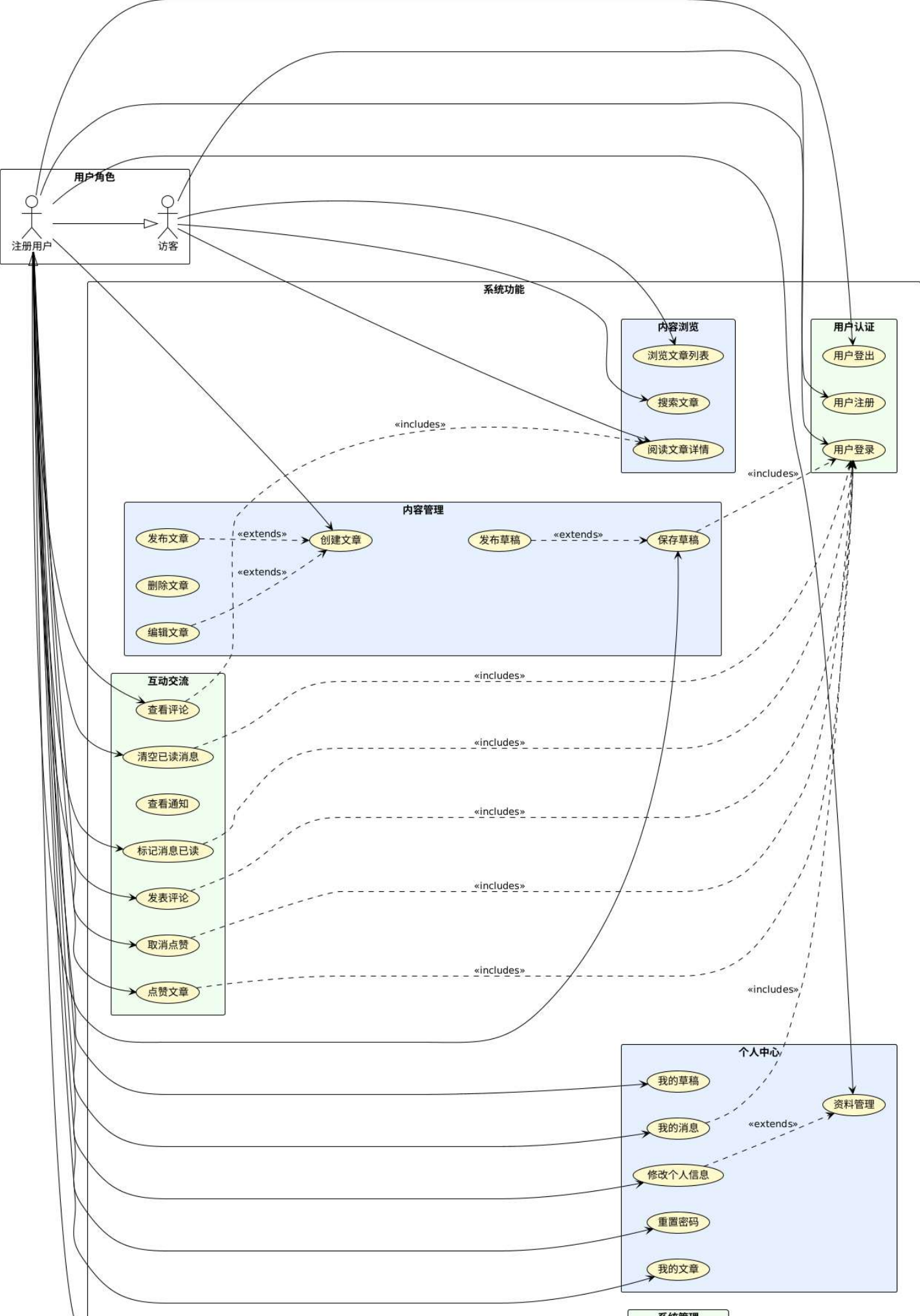
## 2. 系统建模UML图

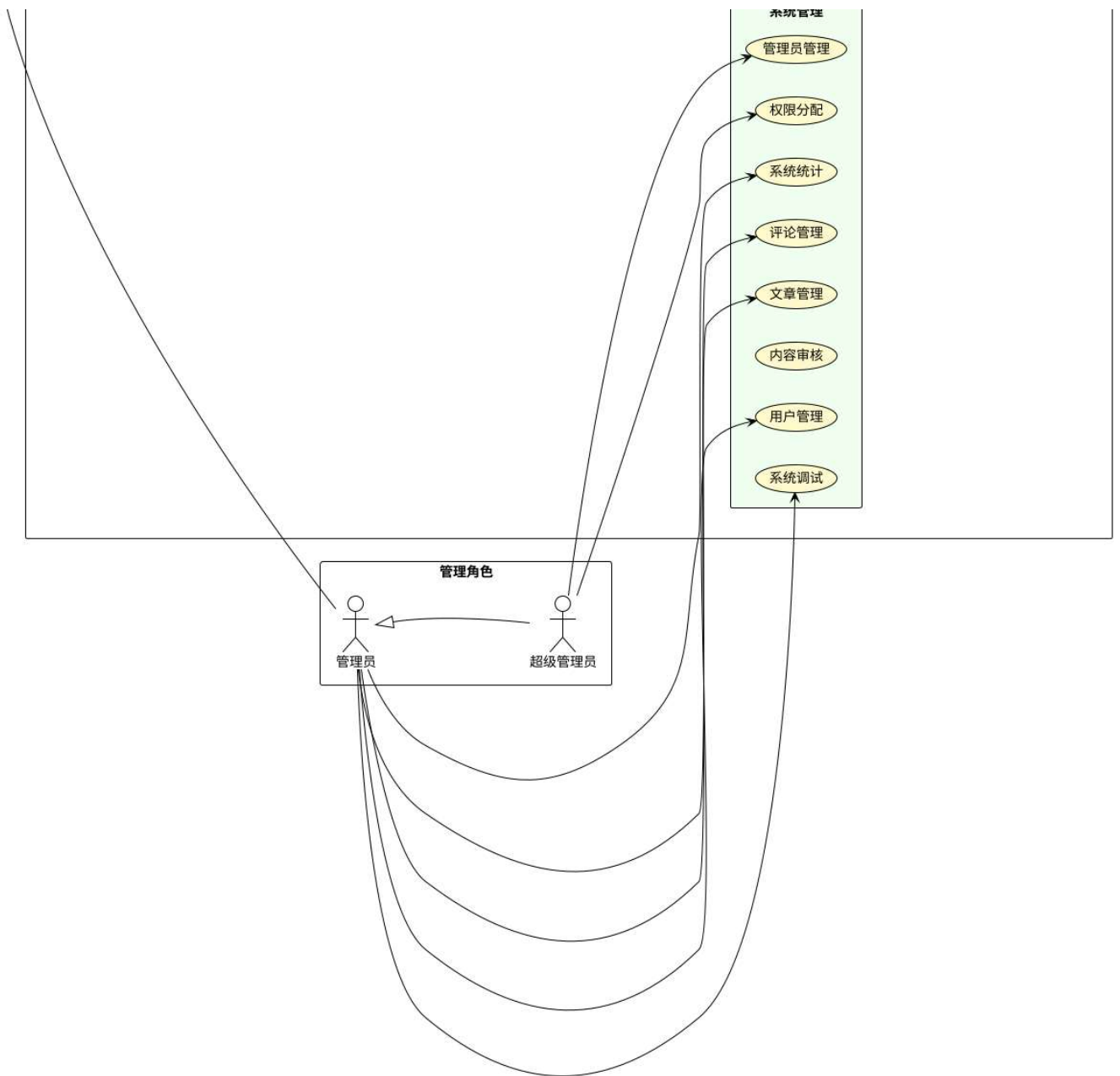
### 2.1 用例图

#### 2.1.1 概述

用例图从用户视角定义了系统的功能边界和主要交互。它识别了与系统交互的“参与者”（Actors）以及他们期望系统提供的服务（“用例”）。该模型是后续所有设计的基础。

Synapse平台用例图





### 2.1.2 参与者 (Actors) 说明

- **访客 (Guest):** 未登录用户，拥有浏览文章、搜索内容等基本权限
- **注册用户 (User):** 系统的核心使用者，继承访客所有权限，并增加了内容创作、评论、点赞等交互功能
- **管理员 (Admin):** 负责系统日常维护，如内容审核、用户管理、数据统计等
- **超级管理员 (SuperAdmin):** 拥有最高权限，负责管理整个系统及其他管理员账户

### 2.1.3 核心用例描述

- **创建文章 (Create Article):** 描述用户从进入编辑器到保存或发布文章的完整流程。此用例通过 `<<extends>>` 关系扩展出"发布文章"和"保存草稿"两个子用例
- **发表评论 (Comment):** 此用例通过 `<<includes>>` 关系依赖于"用户登录"用例，强制要求用户在评论前必须登录，保证了社区环境的规范性
- **内容搜索 (Search):** 系统提供全文搜索、标签筛选等功能，支持访客和注册用户进行内容发现

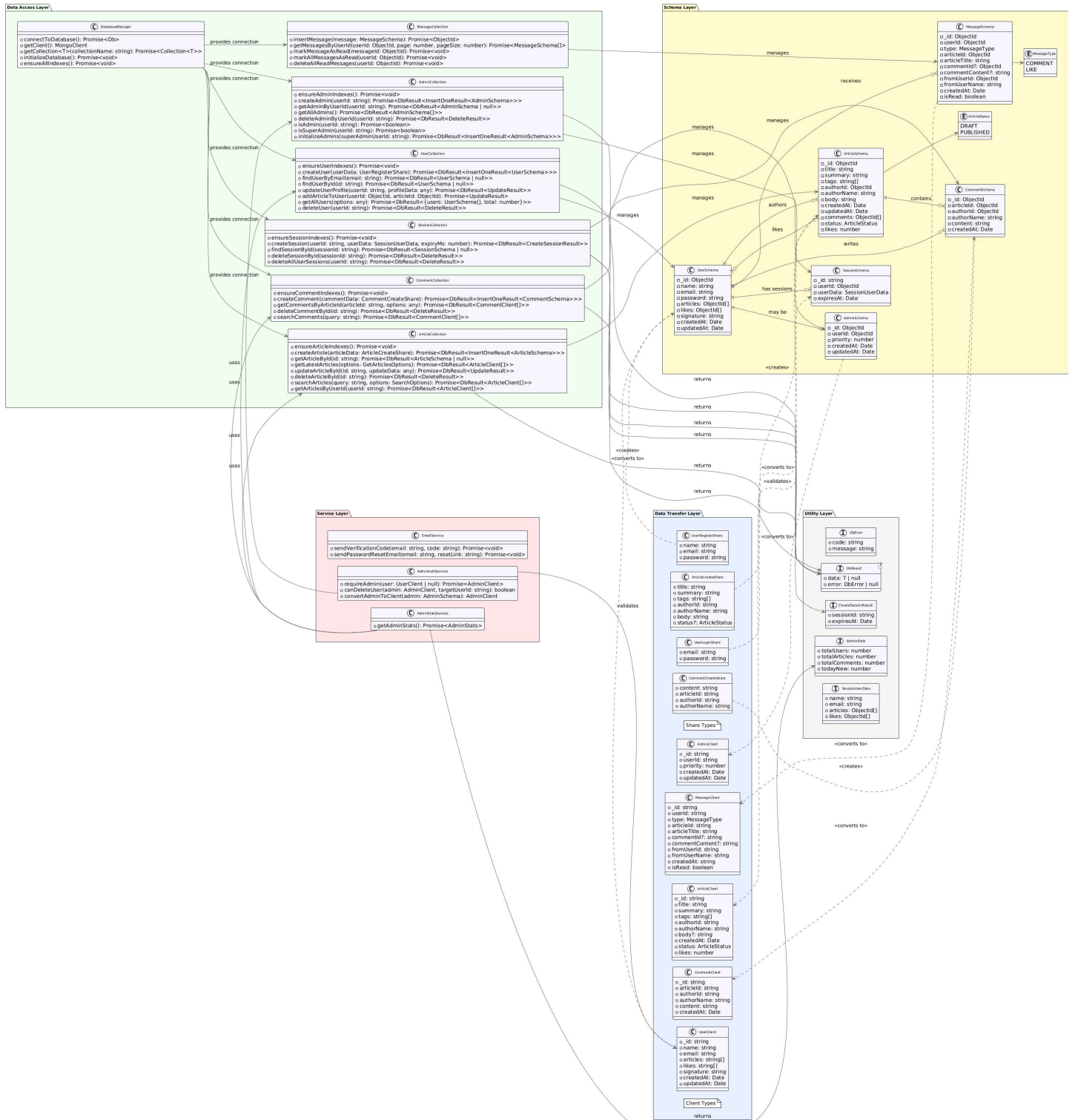
- **管理操作 (Admin Operations):** 管理员可以进行用户管理、内容审核和数据统计，形成完整的平台治理体系

## 2.2 类图

### 2.2.1 概述

类图展示了系统的完整静态结构，包括类、接口、枚举以及它们之间的关系。该图采用分层架构的组织方式，清晰地展现了系统各层次的职责划分和依赖关系。

Synapse平台架构图



## 2.2.2 架构分层说明

### Schema Layer（数据库模型层）

- 包含所有与MongoDB集合对应的TypeScript接口
- 定义了数据的完整结构，包括ObjectId类型和业务属性

- 核心类：UserSchema、ArticleSchema、CommentSchema、MessageSchema、AdminSchema、SessionSchema

### Data Access Layer (数据访问层)

- 封装了所有数据库操作逻辑，提供CRUD和业务查询方法
- 每个Collection类对应一个MongoDB集合的操作封装
- 核心类：UserCollection、ArticleCollection、CommentCollection、MessageCollection、AdminCollection、SessionCollection、DatabaseManager

### Service Layer (业务服务层)

- 提供业务逻辑服务，如权限验证、数据统计、邮件发送等
- 核心类：AdminAuthService、AdminStatsService、EmailService

### Data Transfer Layer (数据传输层)

- 定义了前后端数据交互的安全对象和传输对象
- 将数据库ObjectId转换为字符串，确保前端可以安全处理
- 分为Client类型（前端安全对象）和Share类型（传输对象）
- 核心类：UserClient、ArticleClient、CommentClient、MessageClient、AdminClient、UserRegisterShare、UserLoginShare、ArticleCreateShare、CommentCreateShare

### Utility Layer (工具层)

- 提供系统通用的接口定义和类型声明，确保各层之间的数据交互安全和一致
- 定义了统一的数据库操作结果接口和错误处理机制
- 核心接口：DbResult（统一的数据库操作返回格式）、DbError（错误信息标准化）、SessionUserData（会话用户数据缓存）、CreateSessionResult（会话创建结果）、AdminStats（管理员统计数据）

## 2.2.3 关键设计模式

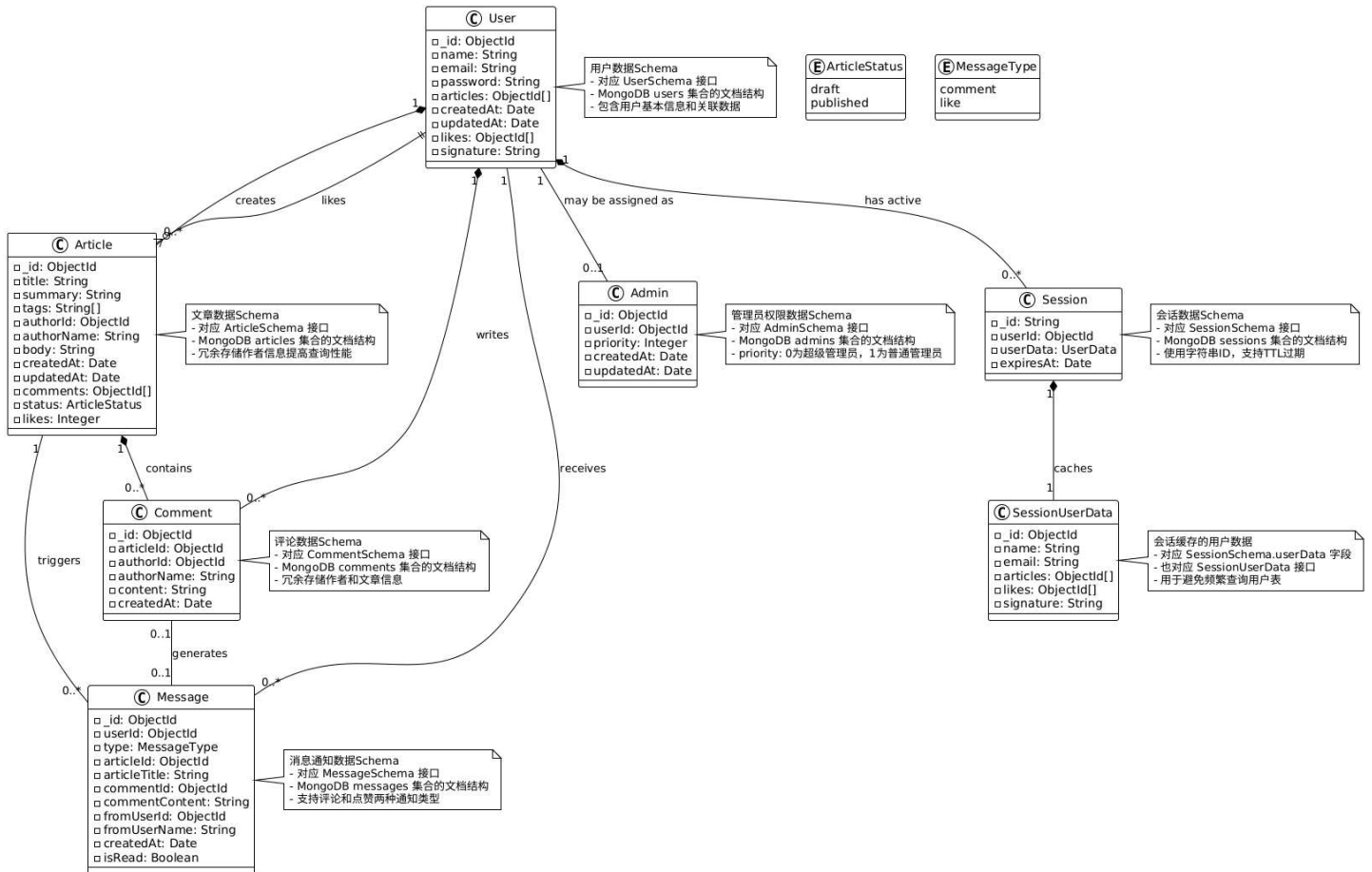
- **Repository模式**: Collection类封装了数据访问逻辑
- **DTO模式**: 不同层次使用不同的数据传输对象
- **Service模式**: 业务逻辑封装在Service类中
- **分层架构**: 明确的层次划分和依赖方向

## 2.3 领域模型

### 2.3.1 概述

领域模型描述了系统的静态结构，定义了系统中的核心实体、它们的属性以及彼此之间的关系。它构成了系统的数据模型和业务逻辑骨架，与实际的TypeScript Schema完全对应。

## Synapse平台领域模型



### 2.3.2 核心实体说明

- **User 实体:** 封装了用户的基本信息（用户名、邮箱、密码哈希）、关联数据（文章列表、点赞列表）和个性化设置（个人签名）
- **Article 实体:** 系统的核心内容实体，包含了标题、摘要、标签、正文、状态（草稿/已发布）、作者信息等完整属性
- **Comment 实体:** 代表用户发表的评论，与 **User** 和 **Article** 实体存在明确的关联关系
- **Message 实体:** 系统消息通知实体，支持评论通知和点赞通知两种类型
- **Admin 实体:** 管理员权限实体，通过priority字段区分普通管理员和超级管理员
- **Session 实体:** 用户会话实体，包含SessionUserData缓存以优化查询性能

### 2.3.3 关键关系与设计决策

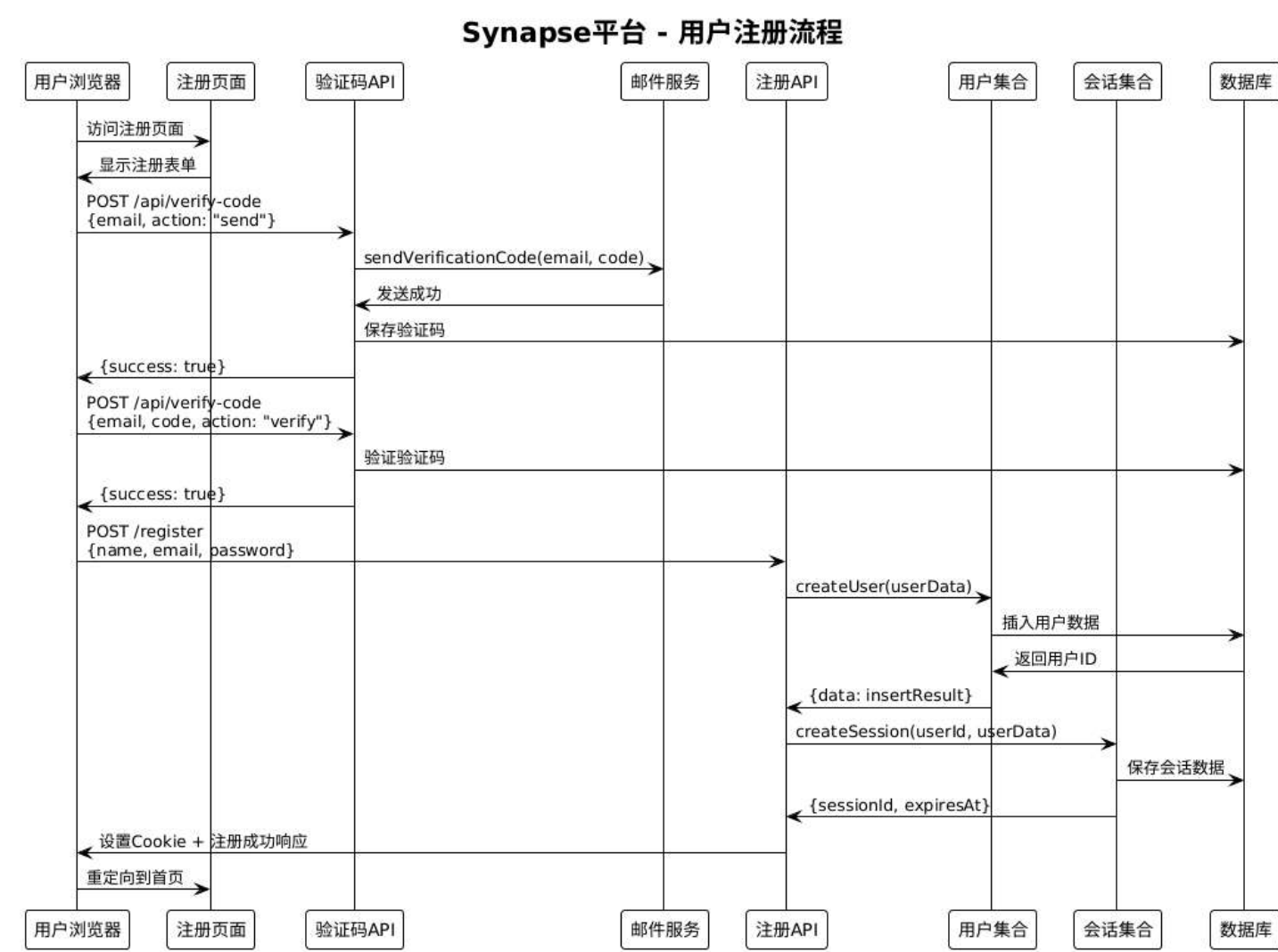
- **User 与 Article 的关系:** 定义为一对多（1:\*）的组合关系。一个用户可以发表多篇文章，文章的生命周期依赖于作者用户的存在
- **Article 与 Comment 的关系:** 定义为一对多（1:\*）的组合关系。评论属于特定文章，如果文章被删除，其下的所有评论也应随之被删除
- **User 与 Admin 的关系:** 定义为一对一的可选关联关系。普通用户可以被提升为管理员，管理员权限独立于用户基本信息进行管理
- **Session 与 SessionUserData 的关系:** 定义为一对一的组合关系，会话缓存用户数据以避免频繁查询用户表

## 2.4 序列图

### 2.4.1 概述

为了更清晰地展示系统在关键业务场景下的对象交互，我们创建了多个模块化的序列图。每个序列图都描述了一个具体用例中系统组件之间消息传递的时间顺序。

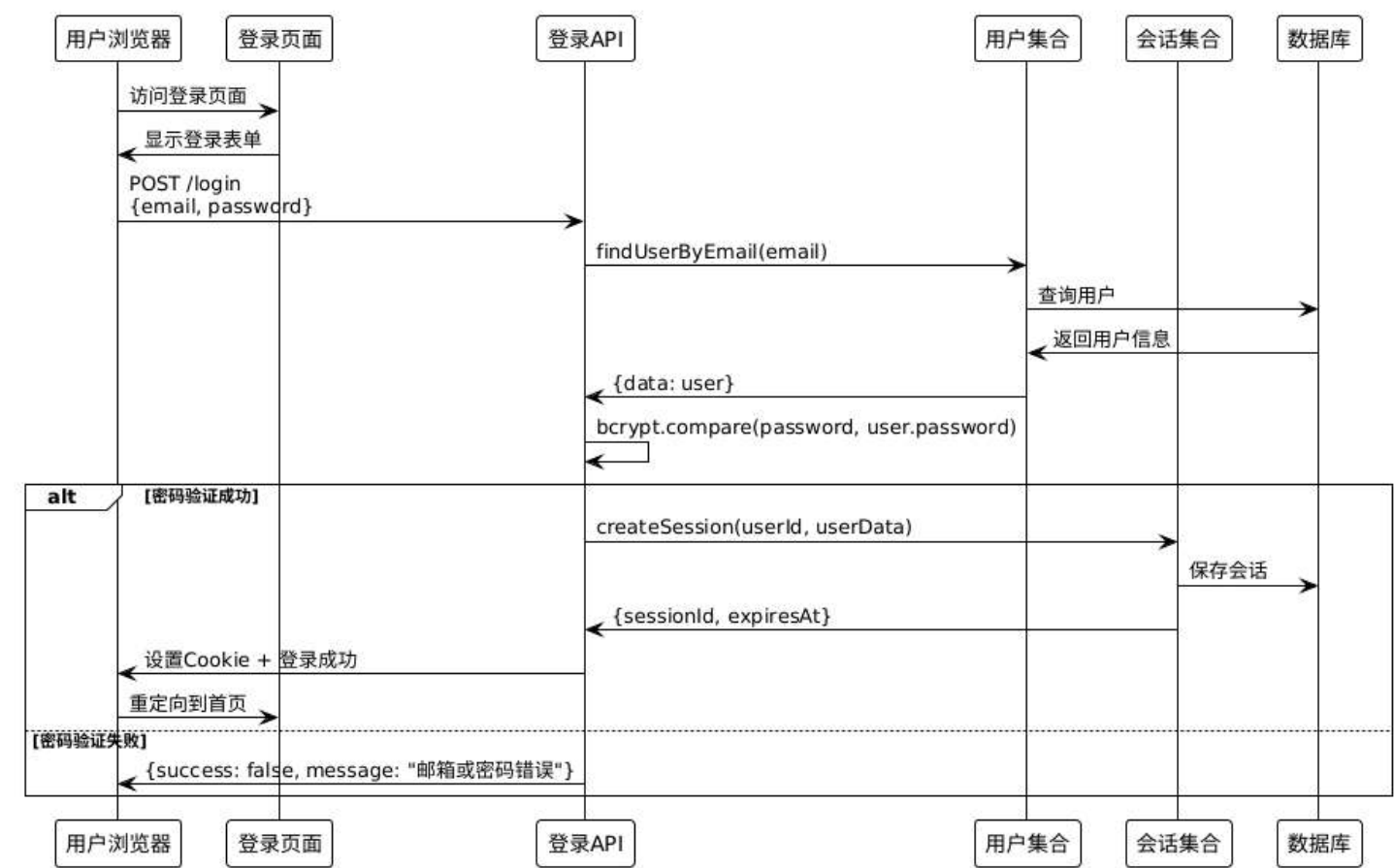
### 2.4.2 场景一：用户注册流程





2.4.3 场景二：用户登录与会话管理

Synapse平台 - 用户登录流程

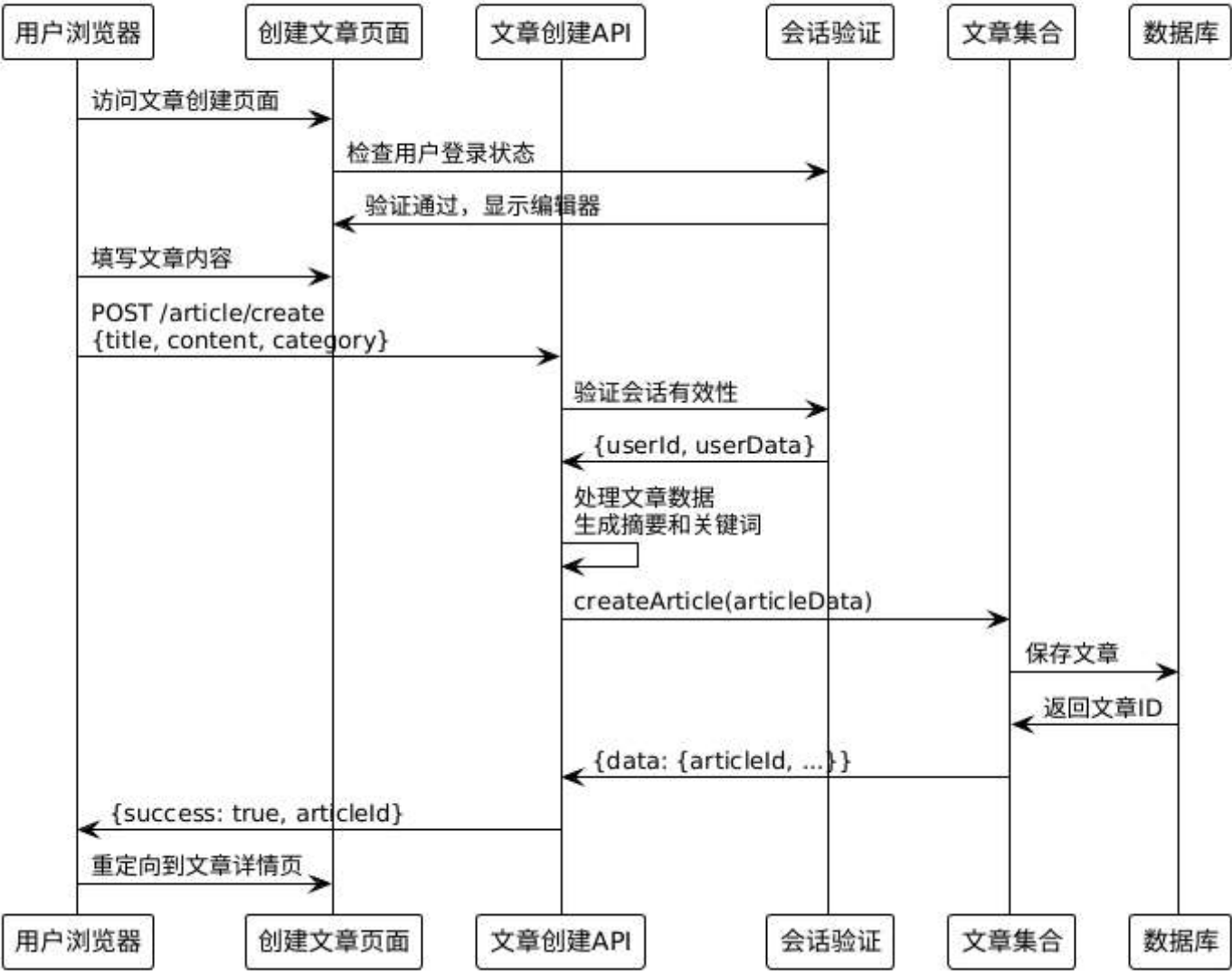


此场景描述了用户登录认证和会话建立的过程，体现了安全的身份验证机制。

**流程描述：** 用户提交登录凭据 → 系统验证密码哈希 → 创建UUID会话 → 设置会话Cookie → 缓存用户数据 → 返回登录成功

2.4.4 场景三：文章创建与发布

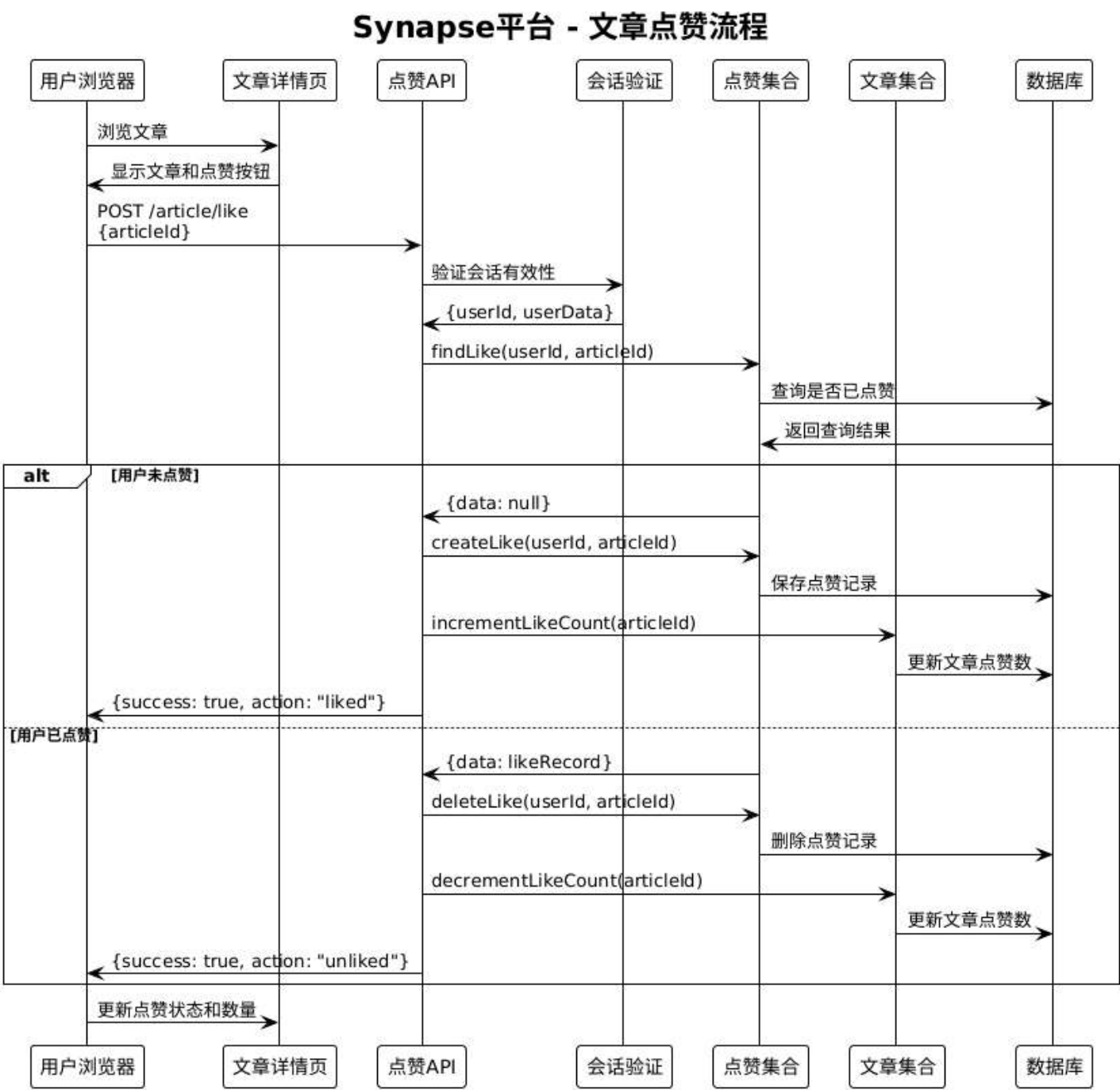
Synapse平台 - 文章创建流程



此场景展示了用户创建文章的交互过程，包括草稿保存和正式发布两种路径。

**流程描述：** 用户编写文章 → 选择保存草稿或发布 → 系统验证内容 → 创建Article对象 → 更新用户文章列表 → 返回操作成功

2.4.5 场景四：文章点赞互动

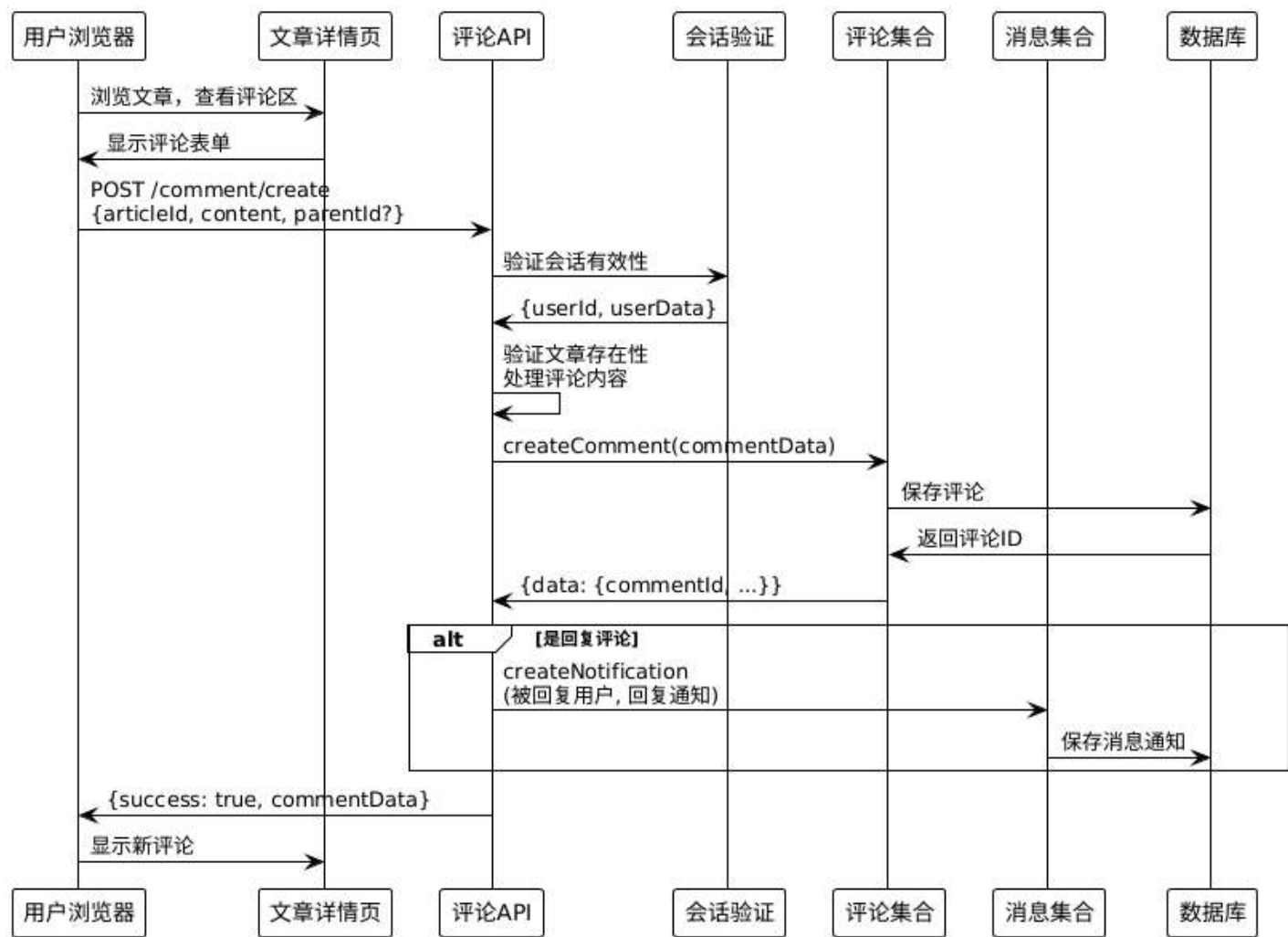


此场景描述了用户对文章进行点赞/取消点赞的交互过程，体现了社交互动的核心机制。

**流程描述：** 用户点击点赞按钮 → 验证登录状态 → 检查是否已点赞 → 创建/删除点赞记录 → 更新文章点赞数 → 返回操作结果

2.4.6 场景五：评论互动流程

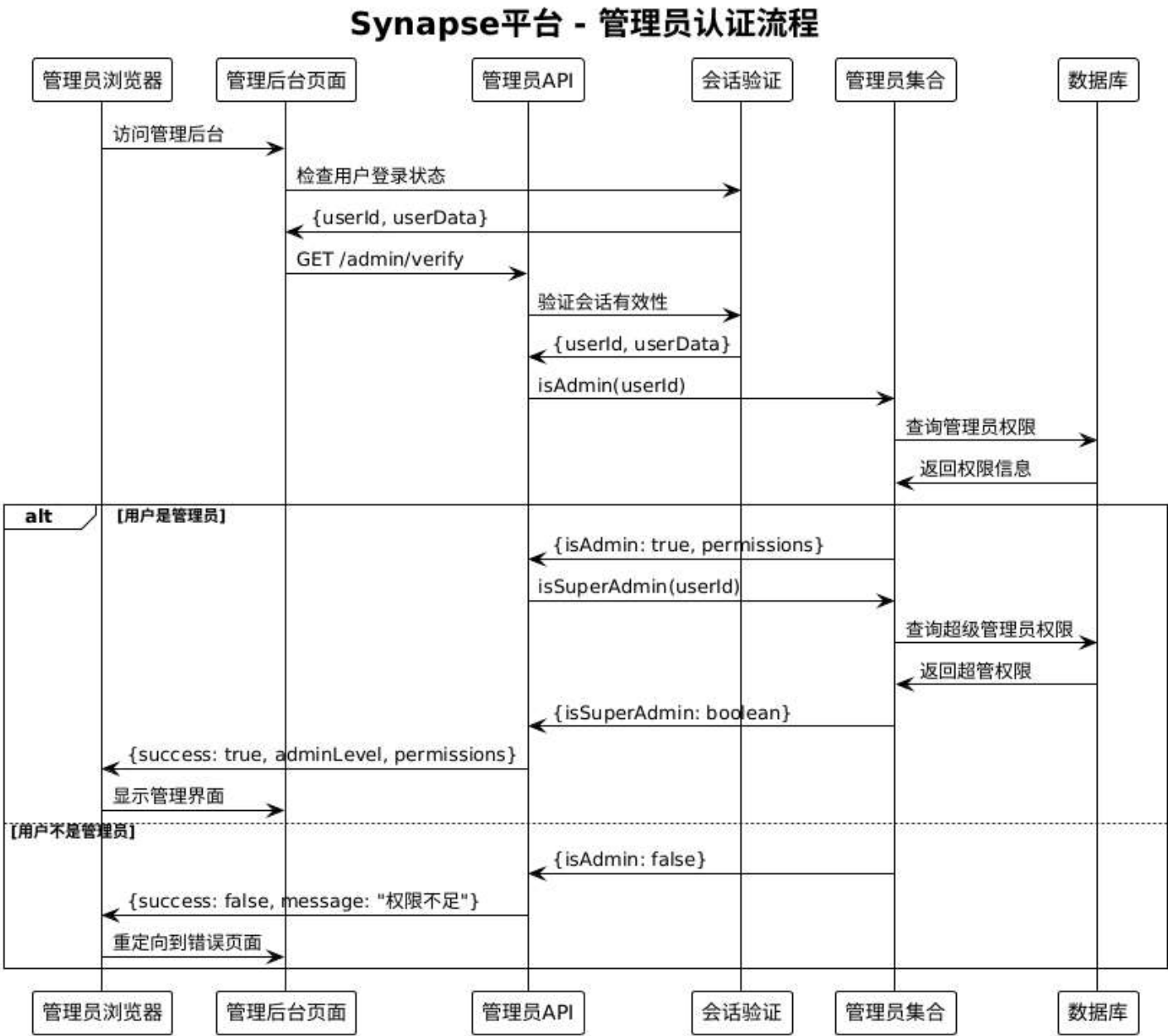
Synapse平台 - 评论创建流程



此场景描述了用户发表评论的交互过程，包括权限验证、消息通知生成等功能。

**流程描述：** 用户提交评论 → 验证登录状态 → 创建Comment对象 → 更新文章评论列表 → 生成消息通知 → 返回评论成功

2.4.7 场景六：管理员权限验证



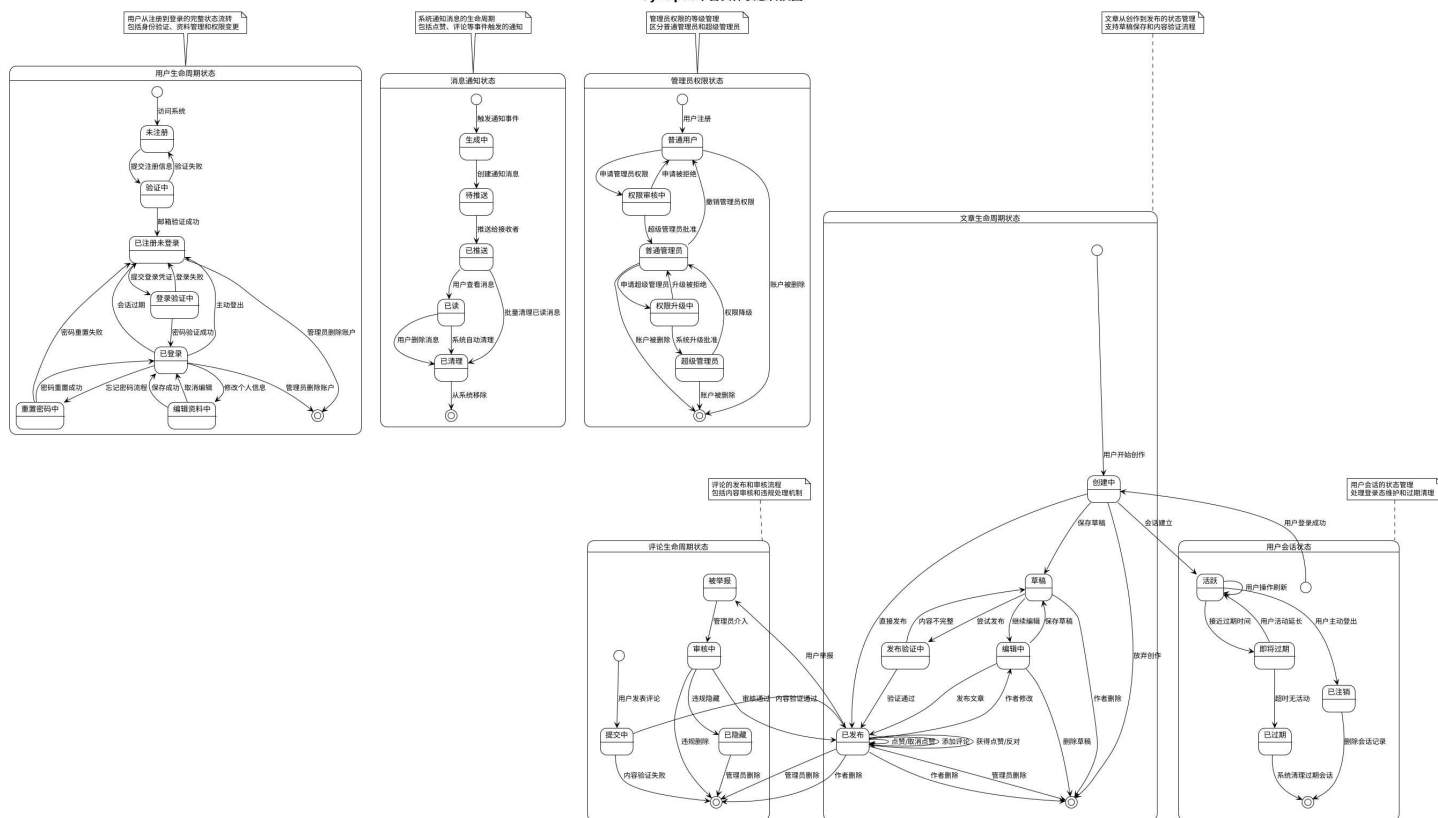
此场景展示了管理员访问后台的权限验证流程，体现了分层权限控制机制。  
**流程描述：** 管理员请求访问 → 验证用户登录 → 查询管理员权限 → 检查权限级别 → 返回访问结果

2.5 状态图

2.5.1 概述

状态图用于描述系统中关键对象在其生命周期内可能经历的各种状态以及触发状态转换的事件。该状态图覆盖了用户、文章、消息通知、用户会话和管理员权限等5个核心实体的完整生命周期。

Synapse平台实体状态转换图



## 2.5.2 核心实体状态生命周期

### 用户生命周期状态

- **状态序列:** 未注册 → 验证中 → 已注册未登录 → 登录验证中 → 已登录 → (编辑资料中/重置密码中) → 已注册未登录/[删除]
- **关键转换:** 邮箱验证成功、密码验证成功、主动登出、会话过期、管理员删除账户
- **设计特点:** 支持邮箱验证流程、密码重置机制、会话管理和管理员干预

### 文章生命周期状态

- **状态序列:** 创建中 → 草稿/已发布 → 编辑中 → 草稿/已发布 → [删除]
- **关键转换:** 保存草稿、直接发布、继续编辑、发布验证、作者/管理员删除
- **设计特点:** 支持草稿机制、发布前验证、已发布文章的互动功能（点赞、评论）

### 消息通知状态

- **状态序列:** 生成中 → 待推送 → 已推送 → 已读 → 已清理 → [移除]
- **关键转换:** 触发通知事件、推送给接收者、用户查看消息、批量清理、系统自动清理
- **设计特点:** 完整的消息生命周期管理，支持自动清理机制

### 用户会话状态

- **状态序列:** 创建中 → 活跃 → 即将过期/已注销 → 已过期/[删除]
- **关键转换:** 会话建立、用户操作刷新、接近过期时间、超时无活动、用户主动登出
- **设计特点:** 智能会话管理，支持活动延长和自动清理

管理员权限状态

- **状态序列:** 普通用户 → 权限审核中 → 普通管理员 → 超级管理员
- **关键转换:** 申请管理员权限、权限审核通过/失败、超级管理员提升、权限撤销
- **设计特点:** 分层权限控制，支持权限申请和审核流程

3. 总结

该系统模型不仅清晰地定义了系统的功能和结构，也为后续的详细设计、开发和测试工作提供了坚实的基础和明确的指导。