|  |  |
| --- | --- |
|  | **计算机与网络空间安全学院** |

**中国旅游推荐系统**

**需求分析报告**

组长：116052020005 叶建行

组员：116052020020 吴一山，

116052020006 林滨，

116052020010 黄志炫

116052020011 杨崇翔，

116052020013 林善雄

**2022年12月6**

目录

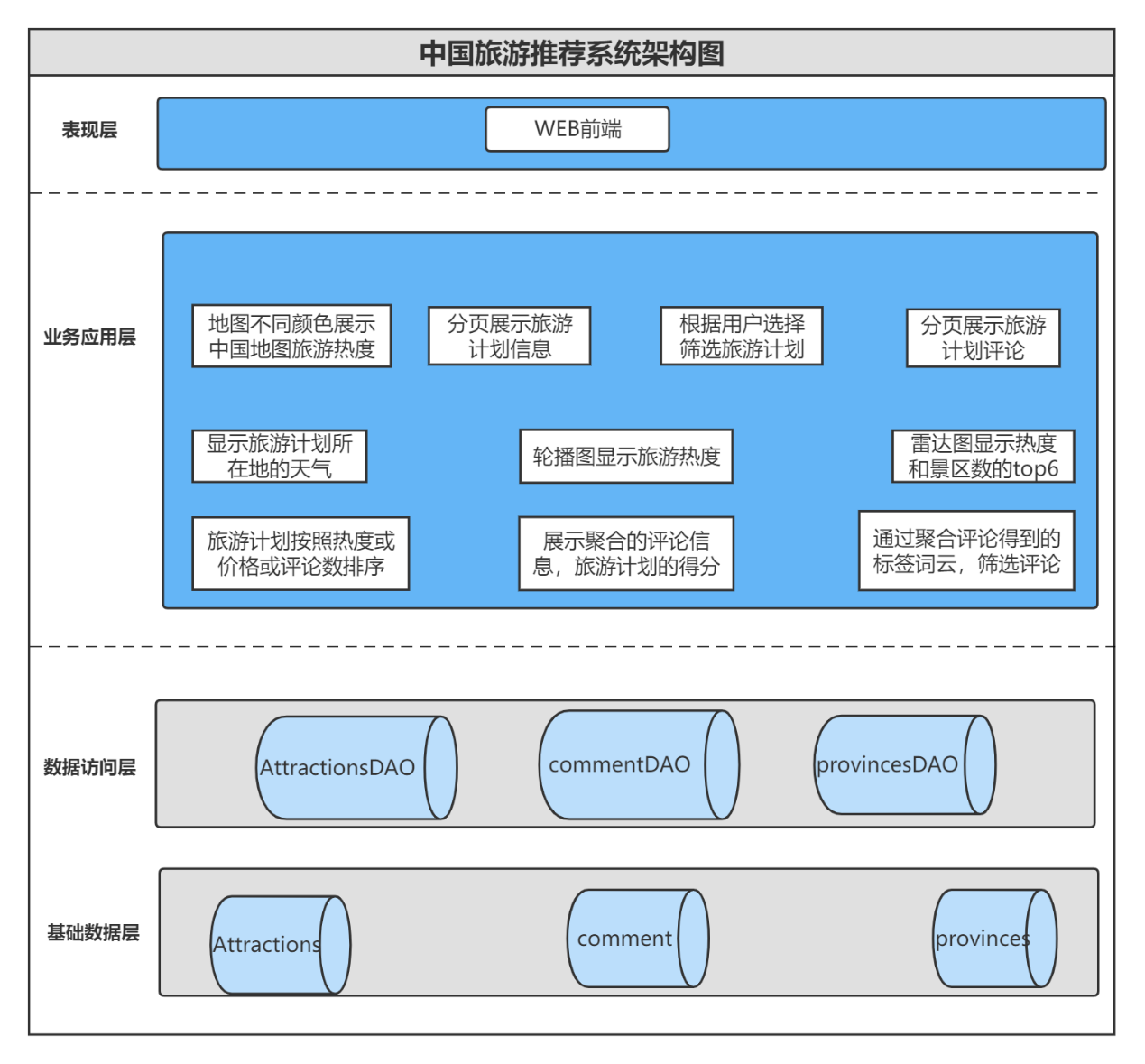
[1 总体设计 3](#_Toc121058385)

[2 数据库概述设计 4](#_Toc121058386)

[3 数据库逻辑设计 7](#_Toc121058387)

[4 数据库物理设计 9](#_Toc121058388)

# 总体设计



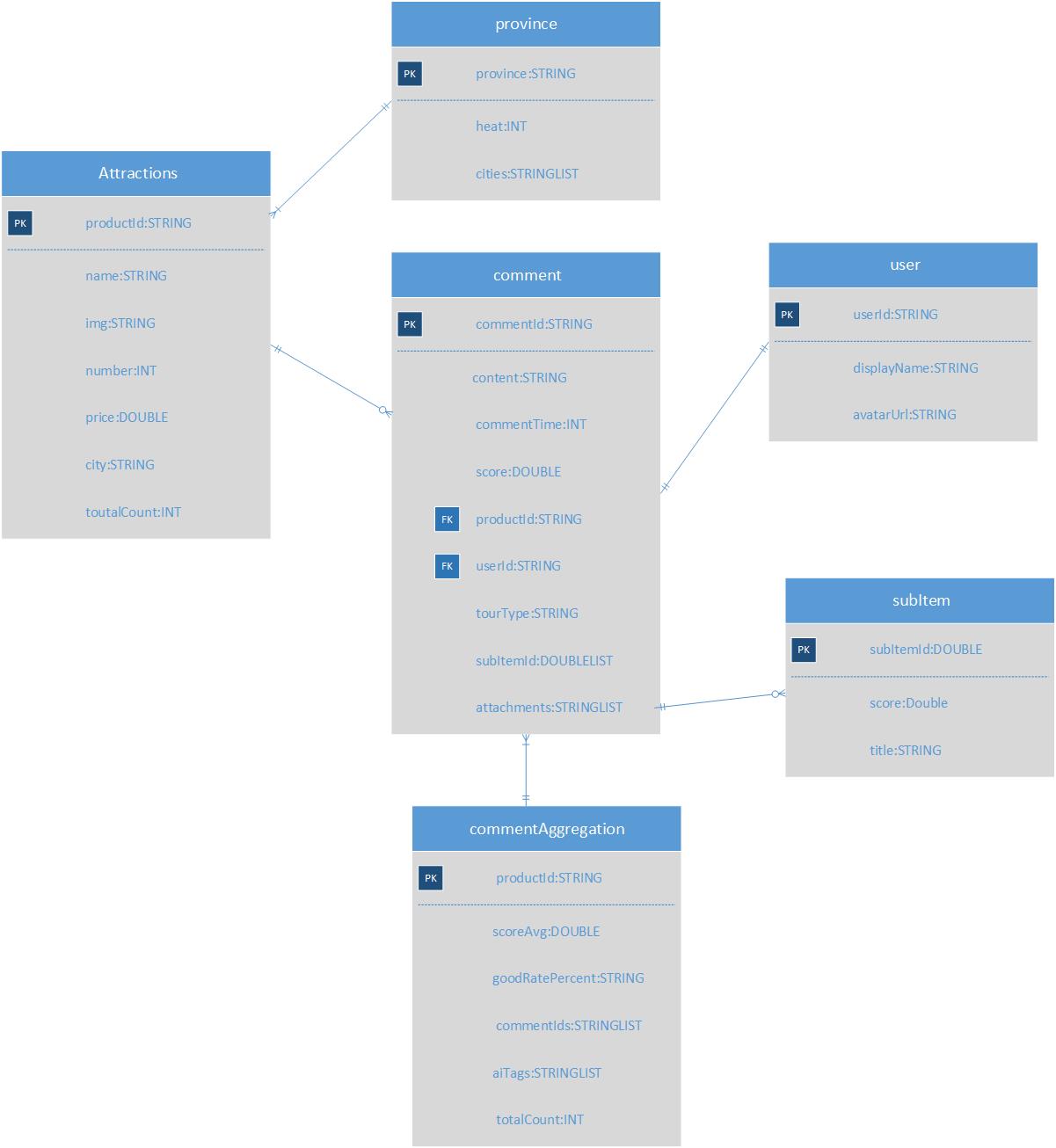
表现层：通过vue技术实现优美的界面，为用户提供良好的交互体验，接受，传递用户的请求，调用业务应用层的接口，展示返回的数据信息

业务应用层：通过spring boot框架实现完善的接口，接收前端发送的请求，访问数据访问层，获得数据，实现特定业务

数据层访问层：操作基础数据库的数据

基础数据库：存储业务数据。

# 数据库概述设计



省份信息表用于记录省份的旅游信息，表中各字段如表2-1所示。

表2-1省份信息表

province

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 描述 | 约束 |
| province | STRING | 省份名称 | 主键 |
| heat | INT | 省份热度 | 缺省值为0 |
| cities | STRINGLIST | 省份下属的市区 | 无 |

旅游计划表用于记录旅游景点的信息，表中各字段如表2-2所示。

表2-2旅游计划表

Attractions

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 描述 | 约束 |
| productId | STRING | 产品id | 主键 |
| name | STRING | 旅游计划名称 | 无 |
| img | STRING | 图片 | 无 |
| number | INT | 旅游人数 | 缺省值为0 |
| price | DOUBLE | 旅游价格 | 缺省值为0 |
| city | STRING | 旅游计划所在市区 | 无 |
| totalCount | INT | 旅游计划评论条数 | 缺省值为0 |

用户信息表用于记录用户信息，表中各字段如表2-3所示。

表2-3用户信息表

user

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 描述 | 约束 |
| userId | STRING | 用户id | 主键 |
| displayName | STRING | 用户昵称 | 无 |
| avatarUrl | STRING | 用户头像 | 无 |

分项表用于记录分项标签信息，表中各字段如表2-4所示。

表2-4分项表

subItem

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 描述 | 约束 |
| subItemId | DOUBLE | 分项id | 主键 |
| score | STRING | 分项得分 | 无 |
| title | STRING | 分项标题 | 无 |

评论表用于记录用户对旅游计划的评论信息，表中各字段如表2-5所示。

表2-5评论表

comment

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 描述 | 约束 |
| commentId | STRING | 评论id | 主键 |
| content | STRING | 评论列表 | 无 |
| commentTime | INT | 评论时间 | 缺省值为0 |
| score | DOUBLE | 评分 | 缺省值为0 |
| productId | STRING | 产品id | 外键 |
| userId | STRING | 用户id | 外键 |
| tourtype | STRING | 旅游计划类型 | 无 |
| subItemId | DOUBLELIST | 分项id | 无 |
| attachments | STRINGLIST | 图片列表 | 无 |

评论聚合表用于聚合评论的信息，表中各自的如表2-6所示

表2-6评论聚合表

commentAggregation

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 描述 | 约束 |
| productId | STRING | 产品id | 主键 |
| scoreAvg | DOUBLE | 用户评分的平均分 | 无 |
| goodRatePercent | STRING | 好评率 | 无 |
| commentIds | STRINGLIST | 评论id列表 | 无 |
| aiTags | STRINGLIST | 智能统计的标签 | 无 |
| toutalCount | INT | 评论数 | 缺省值为0 |

# 数据库逻辑设计

对于采用mongodb的方案，此外给出各个collection的设计。可用json来描述各个collection.

province

{

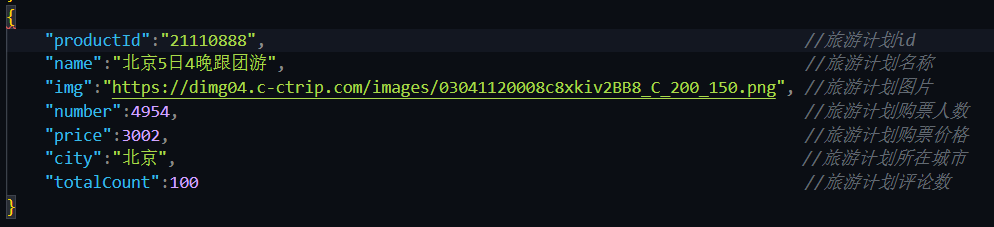
    "province":"北京",                  *//省份名*

    "heat" : 117556,                   *//热度*

    "cities":["朝阳区","海淀区"....]   *//城市名列表*

}

Attractions



comment

（嵌套了comment,user,subitem,commentAggregation四个实体）

对于前提条件：

1. 一对多指向的多端文档数量必须是少量的(one to few),比如<1000个;
2. 整个文档的大小不能超过16MB;
3. 业务上是真正的包含关系,且总是以one作为主体在上下文出现。

考虑到通常情况下用户和评论是一一对应的，且一个评论只有几个subItem,而且评论是这两个实体的主体，所有评论是因为用户产生的，而且一个用户的评论内容不会超过16MB,所以可以嵌套，user,comment,subitem.形成comments,考虑到了comments的数量基本满足one to few的要求，且一个旅游计划的所有评论一般不超过16MB,且commentAggregation依赖于comments所以选择合并comments，commentAggregation，另外这样跟便于评论数据的统计，因为都在一个collection里面，对于一些图片多的评论，会使用懒加载，还有缓存来处理。

{*//某个旅游计划的评论详情和统计信息*

    "productId":"1006894325",                                     *//产品id*

    "comments":[{*//某个旅游计划的评论计划*

        "commentId":"18989573",                         *//评论id*

        "userId":"\_WeCh\*\*\*\*\*\*499699",                   *//用户id*

        "userInfo":{ *//用户信息*

            "displayName":"\_WeCh\*\*\*\*\*\*499699",          *//用户昵称*

            "avatarUrl":"https://dimg04.c-ctrip.com/images/0AS1k120009l60yh36D69.jpg"    *//用户头像*

        },

        "content":"李导讲解得很用心，对团员提出的意见处理得也很及时，处理结果很满意，李导很负责👍🏻。此外，封导也全程默默协助好管理游客的工作👍🏻。", //评论内容

        "commentTime":1627619093811, //评论时间

        "score":5, //评分

        "tourType":"朋友出游", //旅游类型

        "subitems":[{//旅游计划的分项内容

            "subItemId":33997837, //分项id

            "score": 5, //分项得分

            "title":"行程安排" //分项标题

        },{...},{...}],

        "attachments":[ //评论图片

            "https://dimg04.c-ctrip.com/images/300914000000wc4vwBB79\_W\_800\_10000\_Q70.jpg",

            "https://dimg04.c-ctrip.com/images/300j160000010o3r2FB94\_W\_800\_10000\_Q70.jpg"

        ]

    }],

    "totalCount":51, //评论总数

    "commentAggregation":[{//旅游计划评论统计信息

        "scoreAvg":4.9, //所有评论平均分

        "goodRatePercent":"94.1", //好评率

        "subItemTags":[{//旅游计划分项标签统计

            "displayName":"行程安排", //分项名称

            "scoreAvg":4.8, //分项平均得分

            "totalCount":51 //分项数量

        },{...},{...}],

        "tourTypeTags":[{//旅游类型统计信息

            "displayName":"朋友出游", //旅游类型名称

            "totalCount":7 //旅游类型数量

        },{...},{...}],

        "aiTags":[{//智能统计标签信息

            "displayName":"导游服务好", //标签名

            "totalCount":14 //标签数量

        },{...},{...}]

    }]

}

# 数据库物理设计

索引是特殊的数据结构，它以一种易于遍历的形式存储集合数据集的一小部分。索引存储一个或一组特定字段的值，按字段的值排序。索引项的排序支持有效的相等匹配和基于范围的查询操作。此外，MongoDB可以通过使用索引中的排序返回排序后的结果。考虑到系统主要是做查询工作，索引会大大提升查找的效率，因此针对每个collection都设置了索引

provinces定义了province\_1索引，关键词是province,索引中的排序是升序

Attractions定义了name\_1\_city\_1\_price\_1的组合索引索引，关键词是name，city，price,索引中每个字段的排序是升序

comment定义了productId\_1索引，关键词是productId，索引中的排序是升序