

# 功能需求说明和分析v1.2

V1.0: 基本完成初步说明分析

V1.2: 更新图形化界面使用工具; 补充选做功能 4 的“权重”设计

## 1. 游学推荐

- 学生可以根据自己的喜好选择不同的景点和学校作为游学目的地  
**分析:** 景点和学校的 class 里, 需要有不同的标签 (喜好标签), 这些标签将景点和学校分类筛选
- 在游览前, 系统会向学生推荐游学景点和学校, 可以按照游学热度、评价和个人兴趣进行推荐  
**分析:** 景点和学校的 class 里, 需要有热度、评分的属性, 且可随时更改; 将这些 node (一个节点表示一个景点或学校) 的热度值、评分值放在一个表里排序。
- 推荐算法基础要求为排序算法, 可以按照用户选择的热度和评价进行排序;  
(核心算法为**排序算法**, 考虑到用户通常只看前 10 个景点或者学校, 要求不经过完全排序可以排好前 10 的景点或者学校)
- **选做功能 4:** 采用推荐算法 (基于内容推荐、协同过滤推荐) 进行景点、学校、游学日记的推荐  
**分析:** 在图形化界面之前、游学推荐模块和游学日记模块完成之后开始完成。记录用户的搜索、查看、喜欢、感兴趣、交互的景点学校日记 (真正记录的是他们的标签), 根据这些标签的出现频次筛选出前十个景点、学校、日记; 推荐的次序考虑热度+评价+个性化的权重来进行推荐
- 学生可以输入景点和学校的名称、类别、关键字等进行查询, 查询结果有多项时, 可以对查询结果按照热度和评价进行排序。(核心算法为**查找算法**和**排序算法**)  
**分析:** 先进行查询 (字符串比较 (我猜的)), 查询结果进行排序 (提前排好, 在排好的表里查询, 依次输出?)

## 2. 游学路线规划

- 当进入景区或者学校后, 学生可以输入目标景点或者场所信息, 系统会为学生规划从当前位置出发到达景点或者场所的最优游学线路; (核心算法为**最短路径算法**)
- 当进入景区或者学校后, 学生可以输入多个目标景点或者场所信息, 系统会为学生规划从当前位置出发, 参观多个景点或者场所的最优游学线路。(核心算法为**途经多点最短路径算法**, 从当前位置出发, 参观完返回当前位置。)  
**分析:** 维护一个图, 里面有许多 node, 每个 node 都是景点或场所, 有许多条 edge 连接两个 node。给定当前 node 位置, 查询路线。
- 关于线路规划策略的要求:
  - **最短距离策略:** 距离最短即可;  
**分析:** 两个 node 之间最短的 edge 长度和即为所求

- **最短时间策略：** 假设每条道路拥挤度不一样，在这种情况下时间最短即可；拥挤度为小于等于 1 的一个正数，真实速度=拥挤度 × 理想速度；每条道路的拥挤度与理想速度自拟；

**分析：** edge 的属性里，需要添加“拥挤度”的属性（0.0~1.0，数值越大越挤）（拥挤度属性随时间变换，引入时间属性），edge 长度/速度，即为该条 edge 所需时间，求出所需时间最短的 edge 和。

- 交通工具的最短时间策略（**选做功能 3**）：校区内可以选择自行车和步行，选择自行车时，只能走自行车道路，默认自行车在校区任何地点都有；景区内可以选择步行和电瓶车，选择电瓶车时只能走电瓶车路线，电瓶车路线固定，默认上车即走；不同交通工具可以选择时，考虑不同拥挤度的情况下时间最短；

**分析：** edge 的属性里，还需要包括“路的种类”，用交通工具时需要对应这些“路的种类”，然后再采用最短时间策略，选出最快的交通工具。简述：先筛选可用的 edge，然后进行路线规划策略。

- **选做功能 2：** 室内导航策略：模拟教学楼的结构和景区内博物馆等建筑物的内部结构，进行室内导航，包括大门到电梯的导航、楼层间的电梯导航和楼层内到房间的导航；

**分析：** 在路线规划模块全部完成之后再考虑。在一些 node 里，再构建一个图，图里面是一些新的 node（大门电梯等等），根据这些 node 以及他们的 edge 关系，使用路线规划

### 3. 场所查询

- 在景区或者学校内部时，选中某个景点或者场所，会找出附近一定范围内的超市、卫生间等设施，并根据距离进行排序；（**核心算法为排序**）

**分析：** 给定一个 node，查询距离该点路径长度（edge 的和）小于指定距离的所有点；

- 可以通过选择类别对结果进行过滤；

**分析：** 查询到上述的 node 之后，根据 node 的不同属性（场所种类）进行筛选

- 可以由用户输入类别名称查找某个地点附近的服务设施，并根据距离进行排序；（**核心算法为查找和排序**）

**分析：** 完成前两项功能，得到的 node 表进行距离的排序

### 4. 游学日记管理

- 学生游学过程中或者游学结束时可以撰写游学日记，通过文字的方式记录游学内容；

**分析：** 写入文件，包括标题、内容、地点、日期等信息。学生可以编辑、删除自己的游学日记

- 需要对所有学生的游学日记进行统一的管理；

◦ 管理员可以管理所有学生的游学日记，包括审核、删除等操作。

◦ 管理员可以对学生的游学日记进行分类、标签、审核等管理

- 学生可以根据浏览和查询所有学生的游学日记，游学日记的浏览量即为该日记的热度，每位同学浏览完可以对游学日记进行评分；

**分析：** 每浏览一次，热度就++，浏览完评分。评分可通过点赞、星级等方式实现

- 学生在浏览所有游学日记时，可以按照日记热度、评价和个人兴趣进行推荐，推荐算法基础要求为排序算法，可以根据热度和评分进行排序；（核心算法为**排序算法**）

**分析：** 日记的 class 里，具有热度和评分两个属性，根据这两个表排序；根据个人兴趣的排序即为选做功能 4，不多赘述。

- 学生可以输入游学目的地，对目的地相关的游学日记根据热度和评分进行排序；（核心算法为**查找算法**和**排序算法**）

**分析：** 管理每一篇日记的时候，就需要加上该日记的标签属性，该功能先筛选出相关属性的日记，然后排序

- 学生可以输入游学日记的名称进行精确查询；（核心算法为**查找算法**，考虑游学日记数量较大，变化非常快的情况下进行高效查找）
- 可以按日记内容进行全文检索（核心算法为**文本搜索**）
- 可以对游学日记进行压缩存储（核心算法为**无损压缩**）

## 5. 选做功能 1：图形化界面

- 完成上述后使用 Vue&JavaScript 完成

## 6. 选做功能 5：使用 AIGC 算法根据拍摄的景点或者学校的照片进行游学动画生成。

## 游学日记参考



当设计游学日记管理系统时，首先需要明确系统的功能和数据结构，以确保系统能够满足用户需求并且数据能够被有效管理和利用。下面详细讲解系统的功能和存储数据结构的使用：



#### 功能介绍：

##### 1. 学生游学日记撰写和管理：

- 学生可以在系统中撰写游学日记，包括标题、内容、地点、日期等信息。
- 学生可以编辑、删除自己的游学日记。

##### 2. 统一管理：

- 管理员可以管理所有学生的游学日记，包括审核、删除等操作。
- 管理员可以对学生的游学日记进行分类、标签、审核等管理。

##### 3. 浏览和查询：

- 学生和管理员可以浏览和查询所有学生的游学日记，通过关键词、日期、地点等进行检索。

##### 4. 热度统计：

- 系统记录每篇游学日记的浏览量，浏览量即为该日记的热度，用于衡量日记的受欢迎程度。

##### 5. 评分功能：

- 学生可以对浏览的游学日记进行评分，评分可通过点赞、星级等方式实现，用于反映日记的质量和受欢迎程度。

##### 6. 推荐算法：

- 推荐算法基于排序算法，根据日记的热度和评分进行排序。
- 系统可以根据用户的浏览历史、评分历史和个人兴趣，利用协同过滤、内容推荐等技术提高推荐质量。

#### 存储数据结构的使用：

在设计数据库存储结构时，需要考虑到游学日记的各种信息以及系统的需求。下面是可能的数据表和字段：

##### 1. 学生表（Students）：

- 学生ID（StudentID）
- 姓名（Name）
- 学号（StudentNumber）
- 密码（Password）
- 班级（Class）
- ...

##### 2. 游学日记表（TravelDiaries）：

- 日记ID（DiaryID）
- 学生ID（StudentID）（外键，关联学生表）
- 标题（Title）
- 内容（Content）
- 地点（Location）
- 日期（Date）
- 热度（Popularity）
- ...

##### 3. 评分表（Ratings）：

- 评分ID（RatingID）
- 日记ID（DiaryID）（外键，关联游学日记表）
- 学生ID（StudentID）（外键，关联学生表）
- 评分（Rating）
- ...

##### 4. 推荐历史表（Recommendations）：

- 推荐ID（RecommendationID）
- 用户ID（UserID）
- 推荐日记ID（DiaryID）

- 推荐时间 (RecommendationTime)
- ...

#### 5. 标签表 (Tags) :

- 标签ID (TagID)
- 标签名称 (TagName)
- ...

通过以上的数据表设计,可以实现对学生、游学日记、评分、推荐历史等信息的存储和管理。数据库的选型可以根据系统的规模 and 性能需求选择,常见的关系型数据库如MySQL、PostgreSQL以及非关系型数据库如MongoDB都可以胜任这样的任务。

**S** 什么是基于内容推荐、协同过滤推荐算法



基于内容推荐和协同过滤是两种常见的推荐算法,用于向用户推荐他们可能感兴趣的内容或物品。下面分别介绍这两种算法:



**基于内容推荐算法:**