7月14日

简单的需求分析

用例

数据库设计

# 需求分析

1. 核心需求

显式：

游客对已发布的地图(Maps)的**浏览(Search)**和**查看(View)。**（就是说游客能检索public的地图并且点进去看，但是只有看地图的权限）；

游客注册成为用户(Users)；

用户登陆(Login)与退出(Logout)；

用户登陆后获得以下功能：

（在游客功能基础上）

上传图层；(Upload Layer)

新建地图；(New Map)

检索图层；(Search Layer)

在地图能插入图层（以自选类型(Type)）（以默认样式(Style)）；(Insert Layer into Map) //Type指的是分层设色、等级符号、点图、轨迹图等，Style是指图层可视化时的个性化配色和等级分层选项；

保存地图；(Save Map)

编辑已有地图（含权限修改）；(edit map)

编辑已有图层（含权限修改）；(edit layer) //主要是样式的修改

改变底图；(元明清等等的基础行政图)//新增

修改密码；(change password)

隐式：

美观统一的UI界面；

提供一定的系统自带图层（元、清数据）；

绘制图层和地图的功能；(Draw Map with Layers)

图层树等相关控件在地图展示界面的更新；(UI Update)

1. 基本功能需求：

在展示地图界面实现各种小工具：测距、截屏、分享等。

修改并保存图层的样式（随地图保存）；

图层显隐控制；(LayersTree with CheckBox)；

点选查询；(Select by Click)

属性查询；(Select by Attribute)

用户登陆校验；(验证码)

1. 扩展需求：

绘制时间序列图；

用户安全性，非明文密码储存；

提供更多的图层类型；

提供跨地图的图层操作；（比如从一份地图中保留样式将一个图层加载到另一份地图中的功能）

提供给用户向开发人员反应问题的渠道；

提供更为丰富的用户信息（由通用B组负责增加），以及修改这些信息的功能；

给予游客基于已有的public图层新建地图的权限；

提供地理信息扩展查询功能；(Overlay Intersect etc.)；

提供用户对已创建的图层和地图的独立管理界面；

支持用户向服务器端上传CSV数据表，存放，并且在创建图层的时候取出直接使用。

# 用例（基于核心和基本功能需求）

1.基于游客：

点击“查找地图”，进入地图查找和浏览页面，权限是对所有的Public地图，可以搜索地图，也可以直接点击地图，进入地图查看页面。无法对地图进行编辑操作（相关操作功能不出现或者灰色）。

点击注册，输入用户名（需要进行用户名是否重复校验），密码，即可（可以需要输入两次密码，两次密码相同才可以注册）。、

点击登录，输入正确的账号和密码，并且通过验证码校验，即可登录。若输入错误，给出提示，并且停在登录窗口。可以取消正在进行的登录操作，返回游客界面。

2.基于用户：

点击编辑信息，可以修改用户密码，验证当前密码，输入两次想更改的密码，输入验证码，提交，显示修改密码成功的界面后，返回编辑信息界面。

点击“查找地图”，进入地图查找和浏览页面，权限是对所有的Public地图和自己创建的地图，可以搜索地图，也可以直接点击地图，进入地图查看页面。

在地图查看页面，可以做插入图层操作，可以使用已经完成的Public权限的图层，也可以新建图层。若选择新建图层，则进入上传图层数据的界面，在界面当中选择图层的类型，权限等，然后即可直接将默认样式的图层插入到当前地图。

插入图层到地图后，可对图层的样式进行修改，及时渲染修改结果，此后可以执行保存操作，选择地图的权限后，保存时可以覆盖被编辑的地图，也可以另外起一个新的地图（另存为）。

仍然是地图编辑界面，图层的显隐可以再左侧的面板中通过checkbox来修改，叠放顺序，按理说，应当是直接拖动图层的名字来操作。而这些操作，理应是可保存的。同时，对编辑地图界面的操作，提供撤销功能。而对于底图(BaseMap)的操作，则是由单选框完成，提供（无，元，明，清的选项）

用户可以点击工具栏中的截图、分享、测距等按钮，进行相关操作。其中，点击测距后，其他功能应当上锁，待测距完成后解锁。截图功能的用例，需要看可行性分析的结果，暂定为，同样锁定其他功能，鼠标在屏幕上选定截图区域，生成截图，并且保存，然后再推出截图功能，截的图片应当能存在在系统剪贴板上，也可以是和文件下载一样，弹出文件选框，自定保存位置和名字。

用户可以点击点选工具，进入点选模式，然后再界面上点击对应图层的某个Geom，会出现小窗口，显示Geom信息，并且对应的Geom会显示被选定状态。再次点击点选工具退出此状态。

用户可以点击按照属性查询工具，查找符合条件的（往往是含有某个字符串，或者是在某个数字范围内的），Geom，点击查询后，如果没有结果，需要提示，有结果的话，相关结果进入被选定状态。

# 数据库设计

## Users（用户）

存放用户信息，账号，密码，等

|  |  |  |
| --- | --- | --- |
| ID | 自增，主键 |  |
| UserName | Varchar(40) | 不可重复 |
| Password | Varchar(40) | 最好进行加密 |
| Email | Varchar(40) | 扩展信息示例 |

## Maps(地图)

存放地图的权限，绘图信息，所属用户，等；

|  |  |  |
| --- | --- | --- |
| ID | 自增，主键 |  |
| MapName | Varchar(40) |  |
| UserId | Int ,Users外键 | 创建者Id |
| Accessibility | Bool | T表示Public F表示Private |
| CenterX | Float | 中央经度 |
| CenterY | Float | 中央纬度 |
| ZoomLevel | Int | 缩放等级 |
|  |  | 可能还需增加信息 |
| BaseMap | Int | 0:没有  1:元  2:明  3:清  7-14新增，存储地图的底图信息 |

## Layers（图层）

存放图层信息。

|  |  |  |
| --- | --- | --- |
| Id | 自增，主键 |  |
| LayerName | Varchar(40) |  |
| UserId | Users外键，用户id | 所属用户Id |
| CSVsrc | Varchar(40) | CSV文件路径 |
| Accessibility | Int | T public F private |
| Type | Int | 0：分层设色图  1：等级符号图  2：点图  3：轨迹图 |

## MapsLayers（描述地图和图层的NtoN关系）

|  |  |  |
| --- | --- | --- |
| Id | 自增，主键 |  |
| MapId | 地图Id，外键 |  |
| LayerId | 图层Id，外键 |  |

## MapLayerStyle（某个Map的某个Layer的Style）

|  |  |  |
| --- | --- | --- |
| Id | 自增，主键 |  |
| MapLayerId | MapsLayers外键 |  |
| Style | Text | 暂定是json长文本 |

## AncientLocation （我们后台用的古代地名经纬度匹配表）

|  |  |  |
| --- | --- | --- |
| Id | 自增，主键 |  |
| AncientName | Varchar(40) not NULL |  |
| Dynasty | Varchar(40) not NULL |  |
| X | Float not NULL |  |
| Y | Float not NULL |  |

## BaseMaps(自行组织底图信息，便于BaiduMap的OverLay绘制即可，可单表，亦可多表)

|  |  |  |
| --- | --- | --- |
| 举个例子 |  |  |
| Gid | Id 自增 主键 |  |
| DaoName | Varchar(10) | 道 |
| FuName | Varchar(10) | 府 |
| XianName | Varchar(10) | 县 |
| Geometry | Text | Polygon or Point or PolyLine 信息 GeomInfo |
| Etc |  |  |