

深圳市地铁出行管理系统

——基于ArcEngine与C#开发

周财霖 李恒利 李永康 李蓬生 林心怡

目录

Overview

- 1 选题背景和意义
- 2 数据库设计
- 3 关键功能设计
- 4 总结和展望

1

选题背景和意义

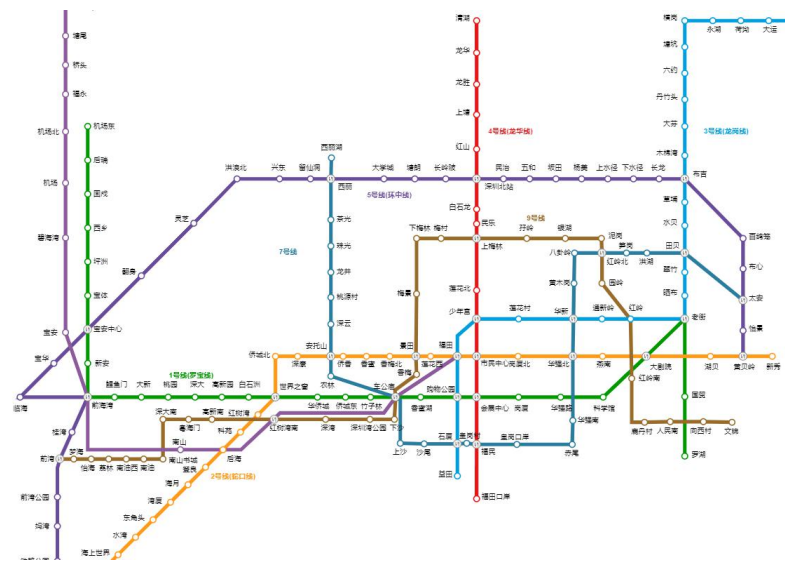
- 研究背景
- 场景需求

选题背景和意义

社会背景

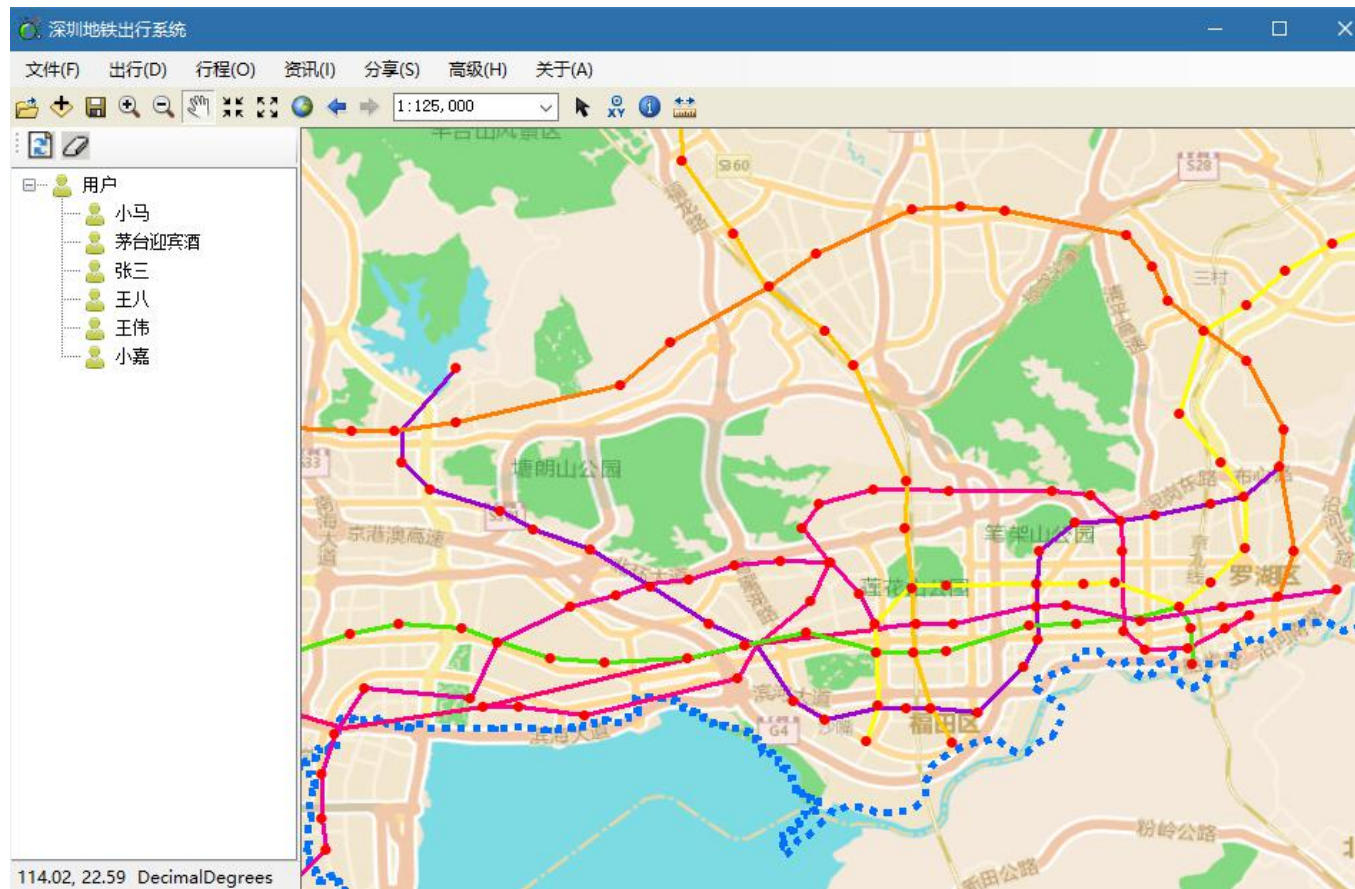
深圳地铁服务于中国广东省深圳市的城市轨道交通。据2019年12月深圳地铁官网信息显示，深圳地铁已开通运营线路共有8条。全市地铁运营线路总长303.44公里，构成覆盖深圳市罗湖区、福田区、南山区、宝安区等六个区的城市轨道网络。地铁出行已经成为深圳市民不可获取的公共交通系统。

目前在移动端有深圳地铁APP，Web端有百度地图等，但在PC端缺少桌面导航和行程记录的出行系统。致力于创建一个桌面应用，实现出行导航，行程记录，社交分享和资讯获取的功能集成，给用户带来完整的深圳地铁出行服务，提升用户的出行和旅途体验，密切地理信息技术与社会活动和日常生活的联系。



功能需求

- 地铁导航
- 行程记录
- 出行信息
- 社交分享
- 路线可视化



2

研究方法和思路

- 技术概览
- 系统结构

技术概览

实现方式

开发语言

- C# (Visual Studio)
- JavaScript

数据库

- PostgreSQL
- Mysql

操作系统

- Windows 10

所用技术

外部库

- ArcEngine
- NewtonJS
- HttpRequest

网络接口

- BaiduMap WebAPI
- map.baidu.com/subway

设计目标

出行导航

- 最短路径查找

路线可视化

- ArcEngine操作

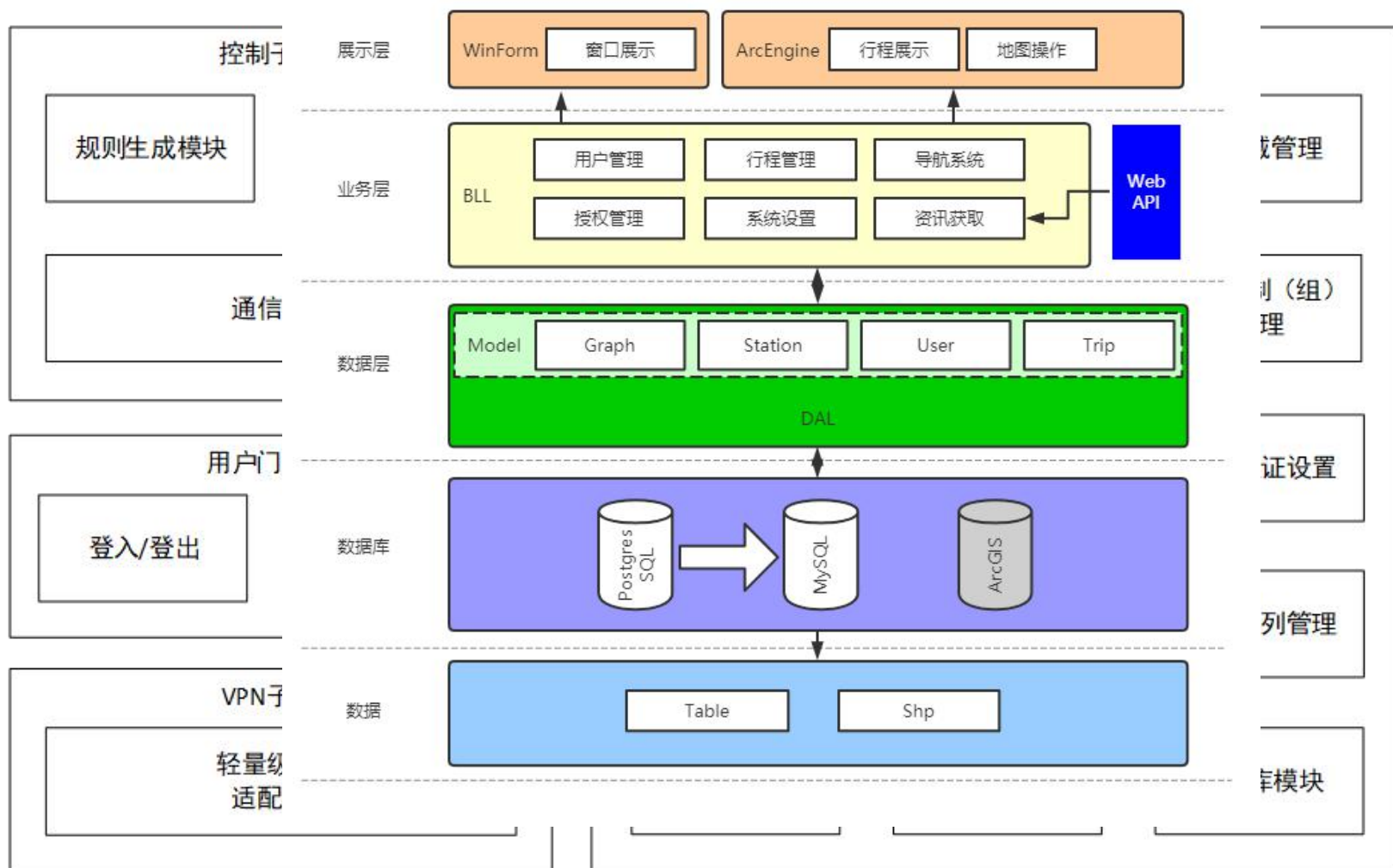
资讯获取

- WebAPI

用户/行程/分享管理

► 研究方法与思路

系统结构图示



3

数据库设计

- PostgreSQL
- MySQL

MySQL

- 关系型数据库管理系统
- 由瑞典MySQL AB 公司开发，属于 Oracle 旗下产品
- MySQL 是最流行的关系型数据库管理系统之一



MySQL优势

- 开源，免费
- MySQL服务器的可扩展，开放的存储引擎
- 支持多线程，充分利用 CPU 资源
- 优化的 SQL查询算法，有效地提高查询速度
- 支持大型的数据库。可以处理拥有上千万条记录的大型数据库
- 使用标准的 SQL数据语言形式
- GIS的空间扩展（5.7新增）

MySQL优势

远程访问

- 云端储存，免去物理拷贝
- 权限控制，维护数据安全

并发访问/分布式访问

- 管理员对数据更新和维护
- 多用户同时使用
- 在多用户大数据情境下，效率远优于Access数据库

```
app.config*  X
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
  </configSections>
  <connectionStrings>
    <add name="MysqlConnection" connectionString="server=localhost; user id=root; password=2333; database=trip;" />
    <add name="Trip.Properties.Settings.MxdPath" connectionString="H:\FILE\sd\final4\final3.0\Data\Map\shenzhen\sz.mxd" />
  </connectionStrings>
  <startup> <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.1"/> </startup> </configuration>
```

```
1 # 创建finley 这只是创建用户并没有权限
2 mysql> CREATE USER 'finley'@'localhost' IDENTIFIED BY 'some_pass';
3 # 把finley 变成管理员用户
4 mysql> GRANT ALL PRIVILEGES ON *.* TO 'finley'@'localhost' WITH
5 GRANT OPTION;
6 #创建用户并赋予RELOAD,PROCESS权限，在所有的库和表上
7 mysql> GRANT RELOAD,PROCESS ON *.* TO 'admin'@'localhost' identified by '123456';
8
9 # 创建keme用户，在test库，temp表，上的id列只有select 权限
10 mysql> grant select(id) on test.temp to keme@'localhost' identified by '123456';
```

评论查询

地铁站名: 华强北

用户	评论
王伟	很开心

查询评论 取消

登录

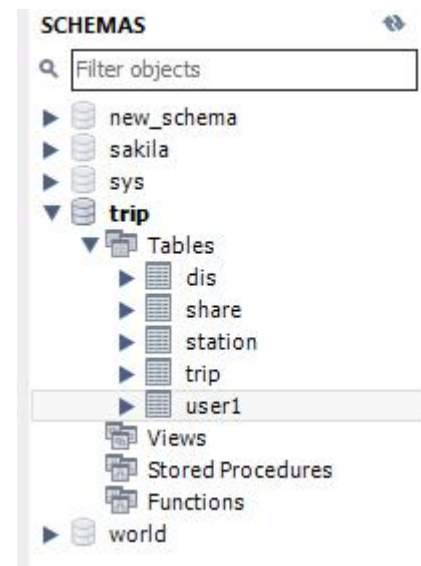
用户名:

密码:

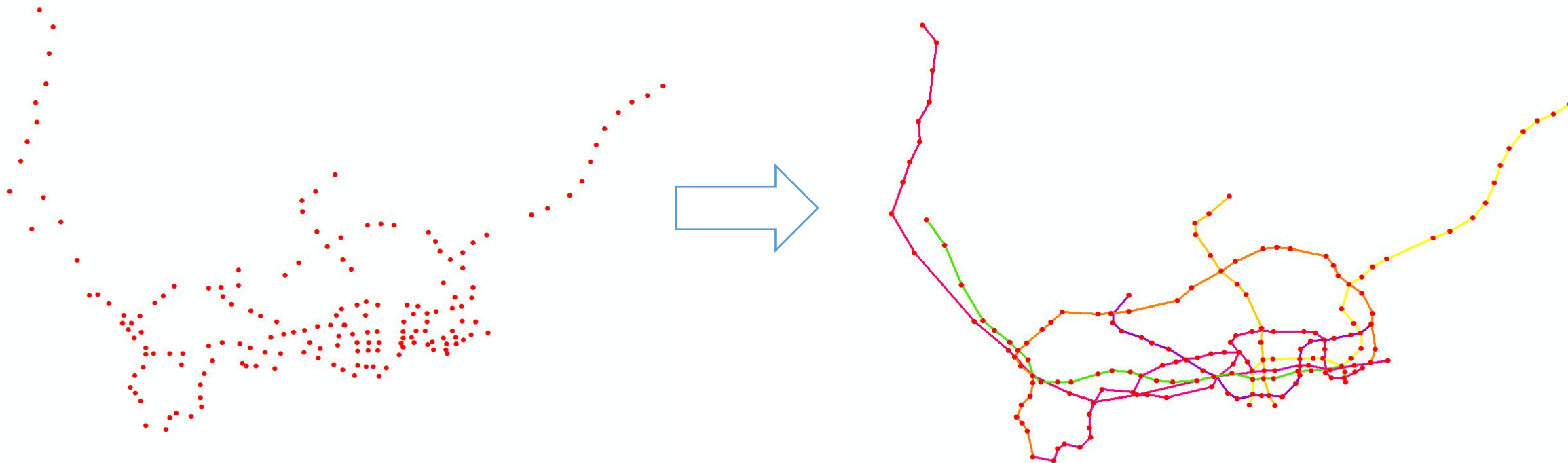
登录 注册 取消

数据库 Trip

- 地铁表 Station (StaID, StaName)
- 用户表 User (UserID, UserName, Sex, Age, Summary, Password)
- 行程表 Trip (TripID, UserID, StaID, StaName, ArriveTime, memo)
- 分享表 Share (ShareID, UserID, UserName, StaName, ShareText)
- 邻站距离表 Distance (DisID, StaID, StaName, LineName, Is_trans, Neigh1ID, Neigh1Name, Dis1, Neigh2ID, Neigh2Name, Dis2, Info)



导航功能 → 最短路径算法 → 创建权重无向图 → 点与点之间的距离 → PostgreSQL ← Shp文件



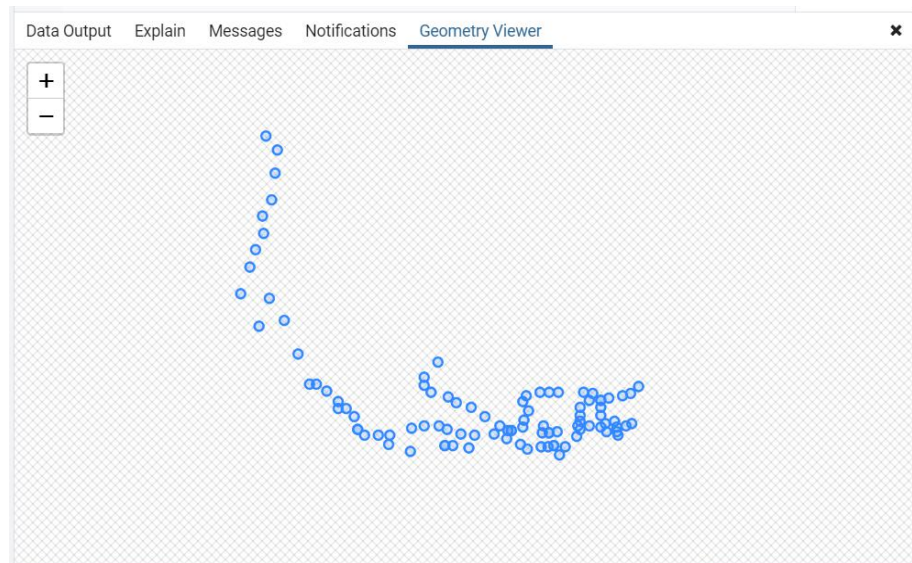
PostgreSQL

- 一种对象-关系型数据库管理系统 (ORDBMS)
- 目前功能最强大、特性最丰富和最复杂的自由软件数据库系统



PostGIS

- 使普通数据库变成空间数据库
- 增加PostgreSQL存储管理空间数据的能力
- 像存储和操作数据库中其他对象一样去存储和操作空间对象
- 多维度空间索引被用于进行空间操作的高效处理



思路

- 构建无向图的关键在于获取每一点的邻接点与边权重
- 通过拓扑关系对线进行切割，计算子线串轨迹长度

```
create view test as select * from import_site;
create view dist_1 as select import_site.fid as ofid,
import_site.邻接点1fid as cfid,
import_site.geom as ogeom,
site_duplicant.geom as cgeom,
ST_Distance(import_site.geom, site_duplicant.geom) as dist from import_site,
site_duplicant where import_site.邻接点1FID=site_duplicant.fid
UPDATE import_site SET 距离1=(select ST_distance(import_site.geom, site_duplicant.geom)
from import_site, site_duplicant where import_site.邻接点1FID=site_duplicant.fid)
update import_site set 距离1=dist_1.dist from dist_1 where dist_1.ofid=fid
```

如何确定站所在的线？

- ST_Relate 确定点与线的拓扑关系



如何确定站与站之间的轨道距离？

- ST_Line_Substring切割获取子串线，ST_Length算距离

如何确定邻站？

- 同一线路上轨道距离最短的站点

	fid [PK] bigint	pointname character varying (20)	neighbor1 character varying (20)	dist1 double precision	neighbor2 character varying (20)	dist2 double precision
1	49	西丽湖	西丽	0.017060866385668196	[null]	[null]
2	50	西丽	茶光	0.006739450404204464	西丽湖	0.006739450404204464
3	51	茶光	珠光	0.008318512019016936	西丽	0.008318512019016936
4	52	珠光	龙井	0.015064992784954387	茶光	0.015064992784954387
5	53	龙井	桃源村	0.00804551768317794	珠光	0.00804551768317794
6	54	桃源村	深云	0.01260289193332694	龙井	0.01260289193332694
7	55	深云	安托山	0.014613840232077461	桃源村	0.014613840232077461
8	56	安托山	农林	0.014400848853378853	深云	0.014400848853378853
9	57	农林	车公庙	0.0109601378178679	安托山	0.0109601378178679

4

关键功能设计

- 用户登录与管理
- 导航与线路可视化
- 行程管理与分享
- web 资讯获取



登录窗口

- 通过已有的用户信息验证用户是否能够登录到应用程序中
- 数据库：User表的UserID与Password字段



A screenshot of a login window titled "登录" (Login). The window has a light blue background with a subtle pattern. It contains two input fields: "用户名:" (Username) and "密码:" (Password). Below the input fields are three buttons: "登录" (Login), "注册" (Register), and "取消" (Cancel).

存储用户注册信息的SQL语句：

```
string sql = "select Name,password from trip.user1 where Name='" + userName + "' and password='" + userPassword + "'";
```



注册窗口

- 通过多个textbox控件获取用户的基本信息并将其存储到用户表中
- 数据库：数据库中User表的Insert操作

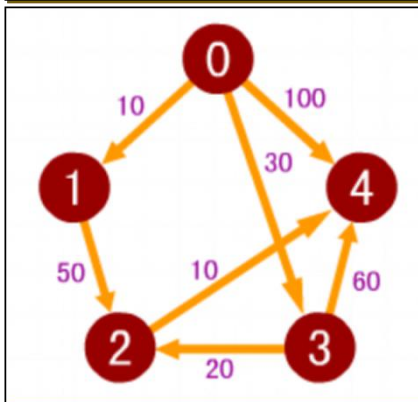
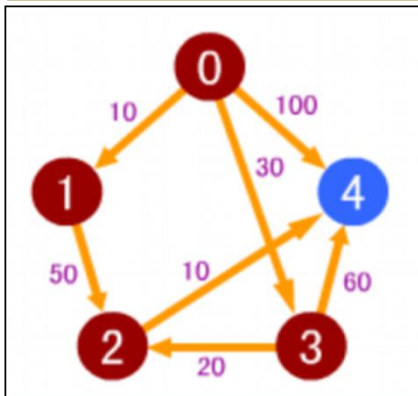
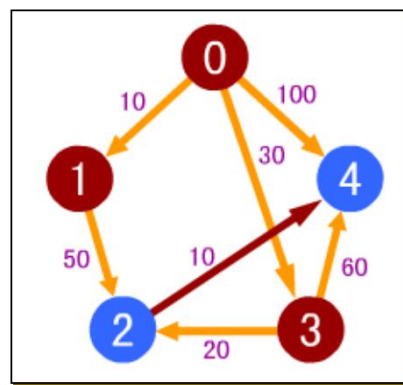
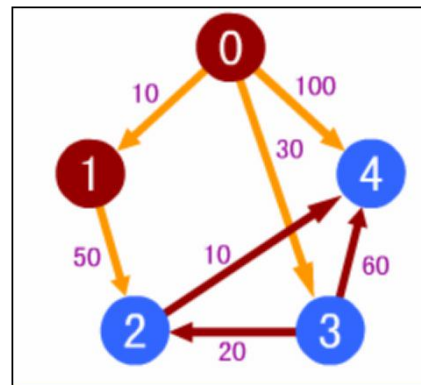
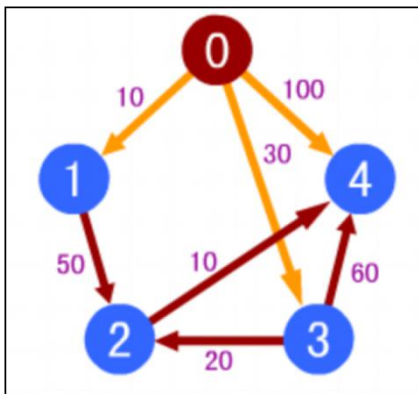
The image shows a 'User Registration' (用户注册) window. It contains several input fields: 'Username' (用户名), 'User ID Card' (用户身份证), 'Gender' (性别) with radio buttons for 'Male' (男) and 'Female' (女), 'Age' (年龄), 'Residence' (居住地), 'Nickname' (备注昵称), 'Password' (密码), and 'Confirm Password' (请确认你的密码). At the bottom, there are 'Confirm' (确定) and 'Cancel' (取消) buttons.

存储用户注册信息的SQL语句

```
string sql = "Insert into trip.user1 VALUES(" + userID + "," + userName + "," + userSex + "," + userAge  
+ ",(Select id from trip.station where name='" + userCity + "'),' " + userSumName + "," + userPassword + "');";
```

迪杰斯特拉最短路径

- 初始化: $S \leftarrow \{v_0\}$;
 $D[i] \leftarrow \text{arc}[0][i], i = 1, 2, \dots, n-1$;
- 求出最短路径的长度:
 $D[j] \leftarrow \min \{ D[i] \}, i \in V-S; S \leftarrow S \cup \{j\}$;
- 修改:
 $D[i] \leftarrow \min \{ D[i], D[j] + \text{arc}[j][i] \}, i \in V-S$
- 判断: 若 $S = V$, 则算法结束, 否则转 2



```

Graph.cs
├── Node
│   ├── Name : int
│   ├── Parent : Node
│   ├── Weight : double
│   └── GetAllWeight() : double
├── Edge
│   ├── Start : int
│   ├── End : int
│   └── Weight : double
└── Graph
    ├── _nodes : List<Node>
    ├── _edges : List<Edge>
    ├── Graph(List<Tuple<int, int, double>>)
    ├── ConvertToEdge(List<Tuple<int, int, double>>) : List<Edge>
    ├── InitNodes() : void
    ├── InitWeights(List<Tuple<int, int, double>>) : void
    ├── GetEdgeByTwoNode(Node, Node) : Edge
    ├── GetNodeByName(int) : Node
    └── Find(int, int, ref double) : Stack
    
```

导航

`void DisRepository.GetWeightAll()`

- 获取Distance表数据，创建Graph类

`Stack DisBLL.Find(startPoint,endPoint,ref dis)`

- 基于算法获取最短路径和距离

`void NaviForm.FillGrid(Stack path)`

- 基于路径堆栈填充表格

`void MainForm.ShowNavi()`

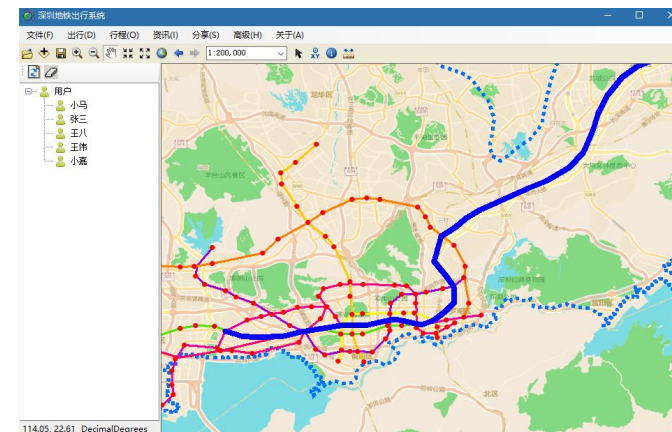
- 创建IElement在地图上显示路径

姓名	线路	换乘提示	资讯
糖坑	3号线(龙岗线)	中间站	龙岗大道(东...)
六约	3号线(龙岗线)	中间站	龙岗大道(南...)
丹竹头	3号线(龙岗线)	中间站	龙岗大道(南...)
大芬	3号线(龙岗线)	中间站	龙岗大道(南...)
木棉湾	3号线(龙岗线)	中间站	龙岗大道(南...)
布吉	3号线(龙岗线)	换乘站	龙岗大道, 龙...
布吉	5号线(环中线)	换乘站	龙岗大道, 龙...
百鸽笼	5号线(环中线)	中间站	信义荔山公园...
布心	5号线(环中线)	中间站	华城百货
太安	7号线	换乘站	深圳市公安局...
太安	5号线(环中线)	换乘站	布心花园、市...
怡景	5号线(环中线)	中间站	荷花市场、峰...
黄贝岭	2号线(蛇口线)	换乘站	新秀立交①
黄贝岭	5号线(环中线)	换乘站	冶金大厦、凤...

查询

途径 11 站

10 元





发表分享功能

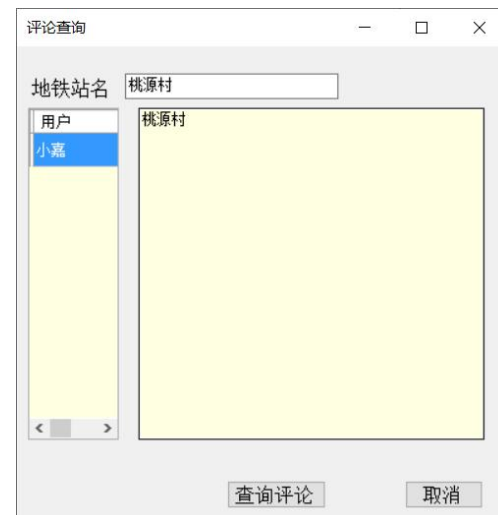
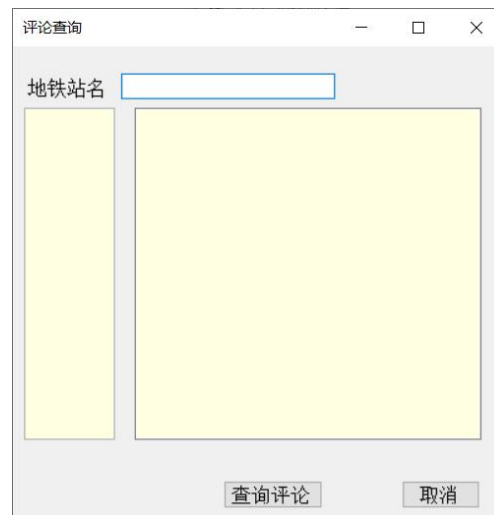
- 通过两个textbox控件获取用户分享的地铁站及其所在的线路，将用户的评论及站点信息存储到分享内容表中
- 数据：数据库中share表的Insert操作

发布分享的SQL语句

```
string sql = "Insert Into trip.share Value(NULL, (SELECT UserID from trip.user1 where Name='" + userName + "'), '" + userName  
+ "', (SELECT CityNo from trip.city where Name='" + stationName  
+ "' and linename='" + lineName + "'), '" + stationName + "', '" + commentContent + "')";
```

查看其他用户分享功能

- 首先根据用户输入的地铁站名，获取已有评论的用户列表，将其显示在DataGridView控件中；
- 然后采用用户单击选取的方式，通过CellMouseClicked事件，获取被选用户的评论



查看分享的SQL语句

```
string sql = "select username,id from trip.share where sitename =' " + stationName + "'";
```

```
string sql = "select sharetext from trip.share where username =' " + getUsername + "' and id=" + getID;
```

行程管理

- 对trip表的select和insert操作

编辑行程

用 户 简介

城 市

时 间

确定 取消

行程管理

添加... 修改... 删除

	用户名	站点	到达时间	备注
*				

上一页 下一页 第 页 共 页 每页 条 关闭



关键功能设计

出行资讯

出行资讯

- 通过webAPI获取资讯信息 (天气WebAPI, 百度地图WebAPI)
- HttpRequest进行url的访问
- get操作进行JSON获取
- 利用NewtonJS扩展中JsonReader对JSON进行解析

```
public InfoForm20
{
    InitializeComponent();
    Timeshow();
    string url = "http://api.avatardata.cn/Weather/Query?key=e83873fc70dd430ab160d3d8e91c019d&cityname=深圳";
    showInfo(url);
}
1 个引用
private string HttpGet(string Url)
{
    HttpRequest request = (HttpRequest)WebRequest.Create(Url);
    request.Proxy = null;
    request.KeepAlive = false;
    request.Method = "GET";
    request.ContentType = "application/json; charset=UTF-8";
    request.AutomaticDecompression = DecompressionMethods.GZip;

    HttpResponse response = (HttpResponse)request.GetResponse();
    Stream myResponseStream = response.GetResponseStream();
    StreamReader myStreamReader = new StreamReader(myResponseStream, Encoding.UTF8); //UTF-8
    string retString = myStreamReader.ReadToEnd();
    myStreamReader.Close();
    myResponseStream.Close();
}
```



5

相关总结和展望

总结

本系统基于C#语言，利用ArcEngine提供地图和地理要素的展示，建立业务逻辑和实体模型进行导航等算法实现，基于MySQL数据库实现相关的数据库操作和交互功能。

展望

- 完善导航算法，得到不同原则下的最短路径
- 加入其他公共交通，例如公交线路，实现多种方式的路径导航
- 利用更多样的webAPI，实现更加丰富的信息获取和资讯展示
- 加入管理员操作界面，实现对数据库和地理要素的同步增改删

6

系统演示
