**iS3 详细设计规约**

ID:GF-DDD

小组成员 1751062 陈恬恬

1752461 刘雨

1754067 徐海琪

修订历史

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 编写日期 | SEPG | 版本 | 说明 | 作者 | 评审时  间 | 参与人  员 | 批准日  期 | 确认人  员 |
| 2015-12-28 | GF | 0.1 | 初稿 | 王泽宇 |  |  |  |  |
| 2015-12-30 | GF | 0.2 | 模块接口 | 王泽宇 |  |  |  |  |
| 2015-01-03 | GF | 0.2.1 | 定稿 | 王泽宇杨安琪 |  |  |  |  |

目录

[修订历史 2](#_bookmark0)

[引言 5](#_bookmark1)

* 1. [编写目的 5](#_bookmark2)
  2. [背景与依据 5](#_bookmark3)
  3. [参考资料 5](#_bookmark4)

[系统软件结构 5](#_bookmark5)

[社交管理模块 5](#_bookmark6)

* 1. [模块描述 5](#_bookmark7)
  2. [功能 5](#_bookmark8)
  3. [模块详细设计 6](#_bookmark9)
     1. [评论图片详细设计 6](#_bookmark10)
     2. [评论小组详细设计 8](#_bookmark11)
     3. [评论专辑详细设计 9](#_bookmark12)
     4. [收藏图片详细设计 10](#_bookmark13)
     5. [收藏专辑详细设计 12](#_bookmark14)
     6. [关注好友详细设计 13](#_bookmark15)
     7. [加入小组详细设计 15](#_bookmark16)

[浏览模块 17](#_bookmark17)

[4.1 模块描述 17](#_bookmark18)

[4.2 功能 17](#_bookmark19)

[个人账户管理模块 18](#_bookmark20)

[5.1 模块描述 18](#_bookmark21)

[5.2 功能 18](#_bookmark22)

* 1. [模块详细设计 18](#_bookmark23)
     1. [登录详细设计 18](#_bookmark24)
     2. [注册详细设计 19](#_bookmark25)
     3. [登出接口详细设计 21](#_bookmark26)
     4. [修改个人信息详细设计 22](#_bookmark27)

[个人主页管理模块 25](#_bookmark28)

[6.1 模块描述 25](#_bookmark29)

[6.2 功能 25](#_bookmark30)

* 1. [接口设计 25](#_bookmark31)
     1. [修改小组信息详细设计 25](#_bookmark32)
     2. [删除小组详细设计 27](#_bookmark33)
     3. [发起小组详细设计 27](#_bookmark34)
     4. [删除专辑详细设计 29](#_bookmark35)

[6.3.5.1 输入项： 29](#_bookmark36)

* + 1. [新建专辑详细设计 29](#_bookmark37)
    2. [修改专辑信息详细设计 31](#_bookmark38)
    3. [上传图片详细设计 32](#_bookmark39)
    4. [修改图片信息详细设计 35](#_bookmark40)

[6.3.9 删除图片信息详细设计 37](#_bookmark41)

引言

1.1 编写目的

《详细设计书》是项目设计的第二个阶段，这个阶段的主要任务是在《概要设计书》基础上， 对概要设计中产生的功能模块进行过程描述，设计功能模块的内部细节，包括类的详细逻辑和接口及类之间的调用关系，为编写源代码提供必要的说明。

本文档的目的旨在推动软件工程的规范化，使设计人员遵循统一的详细设计书写规范，节省制作文档的时间，降低系统实现的风险，做到系统设计资料的规范性与全面性，以利于系统的实现、测 试、维护、版本升级等。

概要设计解决了软件系统总体结构设计的问题，包括整个软件系统的结构、模块划分、模块功能和模块间的联系等。详细设计则要解决如何实现各个模块的内部功能，即模块设计。具体的说，模块设计就是要详细描述各模块中包含的类的处理逻辑及其提供接口的参数格式、请求方式。但这并不等同于系统实现阶段用具体的语言编码，它只是对实现细节作精确的描述，这样编码阶段就可以将详细设计中对功能实现的描述，直接翻译、转化为用某种程序设计语言书写的程序。详细设计的详细程度，应达到可以编写程序的水平。

1.2 背景与依据

iS3平台是地下工程管理平台，可以对地下工程整个生命周期的数据进行采集、管理、可视化和分析。本文档依据《需求分析规约》。

1.3 参考资料

［1］iS3 概要设计规约（说明书）［2016-01－13］

［2］GB8567-88，计算机软件产品开发文件编制指南

系统软件结构

见《需求分析规约》第 2.1 节。

核心类

3.1 描述

核心类存储所有的非图形信息。是MVC设计模式的Model。核心类由Project为基础，存储

所有非图形信息。Project由ProjectDefinition和Domain组成，ProjectDefinition定义项目信

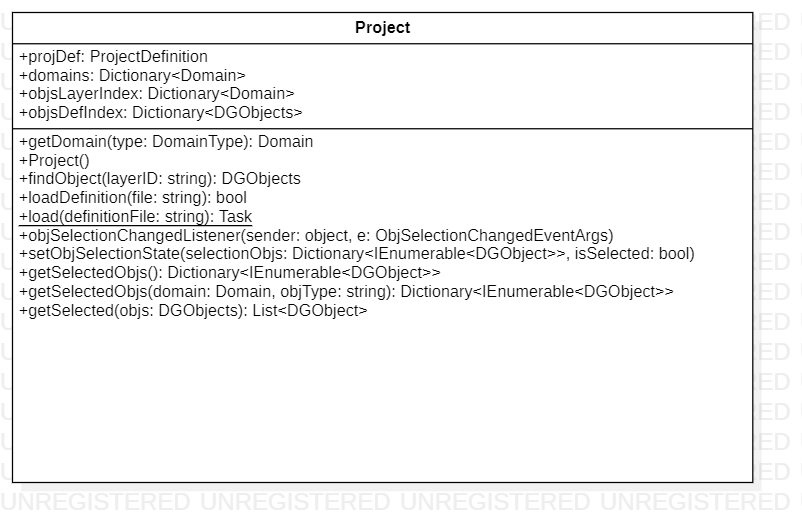
息，所有工程数据存储在Domain中，

3.2 功能

1. 存储数据。
2. 从数据库中加载数据。
3. 从本地文件加载信息。

3.3 类详细设计

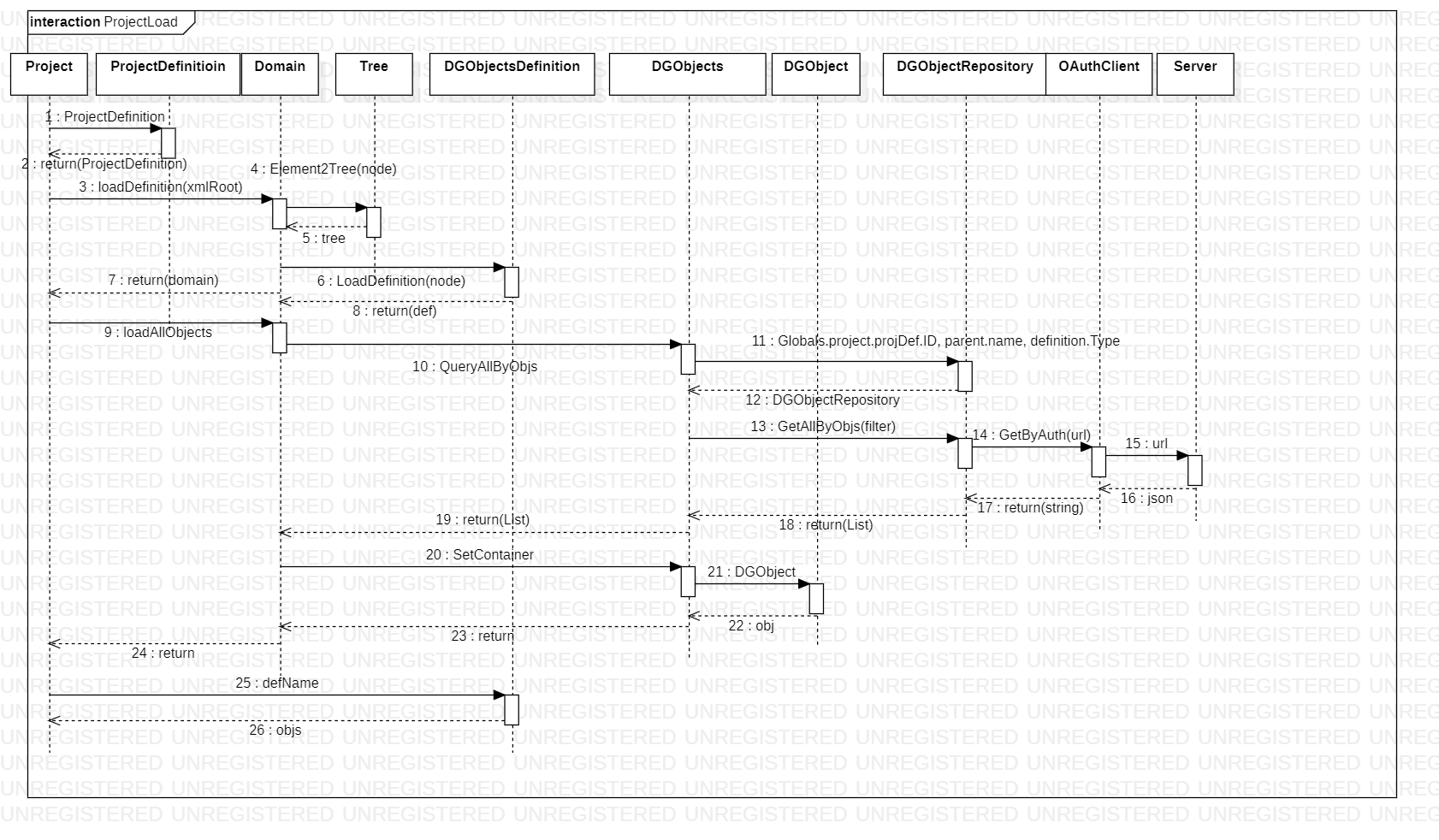
* + 1. Project详细设计
       1. 类图



* + - 1. 属性及方法描述

|  |  |  |  |
| --- | --- | --- | --- |
| Class Project | 变量 | ProjectDefinition projDef | 工程定义 |
| Dictionary<string, Domain> \_domains | 工程 domain 索引 |
| Dictionary<string, DGObjects> objsLayerIndex | 图层名和对象索引 |
| Dictionary<DataSet, DGObjects> dataSetIndex | Dataset 和对象的索引 |
| 方法 | Domain this[string key] | 返回关键字的 domian |
| Domain getDomain(DomainType type) | 返回对应类型的 domain |
| DGObjects findObjects(string layerID) | 找到指定图层上的对象 |
| bool loadDefinition(string file) | 加载工程定义文件 |
| int syncObjectsOnTree  (Tree inputTree, bool readSubTree = true) | 同步数据树和对象，返回  同步类型数 |
| static Project load  (string definitionFile) | 加载 Project |
| void objSelectionChangedListener(object  sender, ObjSelectionChangedEventArgs e) | 对象选取更改事件监听 |
| setObjSelectionState  (Dictionary<string, IEnumerable<DGObject>> selectedObjs,bool isSelected) | 设置指定对象选取状态 |
| Dictionary<string, IEnumerable<DGObject>>  getSelectedObjs() | 获取当前选择的所有对象 |
| Dictionary<string, IEnumerable<DGObject>> getSelectedObjs  (Domain domain, string ObjType) | 获取指定 domain，objType  下的选取对象 |
| List<DGObject> getSelected(DGObjects objs) | 获取对象组中的选取对象 |

* + - 1. 核心方法实现流程



* + - 1. 类内、类外方法调用状态图

待补

* + 1. ProjectDefinition类详细设计
       1. 类图



* + - 1. 属性及方法描述

|  |  |  |  |
| --- | --- | --- | --- |
| class ProjectDefinition | 变量 | string ID | 工程 ID |
| string ProjectTitle | 工程名称 |
| string DefaultMapID | 默认图层 ID |
| string LocalFilePath | 文件目录 |
| string LocalTilePath | TPK 文件目录 |
| string LocalDatabaseName | 数据库文件名称 |
| string DataServiceUrl | 数据服务 URL |
| string GeometryServiceUrl | 地图服务 URL |
| List<ProjectInformation> SubProjectInfos | 子工程信息 |
| List<EngineeringMap> EngineeringMaps | 工程地图索引 |
| 方法 | ProjectInformation GetSubProjectInformation  (string subProjectID) | 获取子工程信息 |
| EngineeringMap GetEMap(string mapID) | 获取相应的工程地图 |
| static ProjectDefinition Load  (XElement root) | 从工程 XML 文件加载工程  定义 |

* + - 1. 核心方法实现流程

同图 3.3.1.3

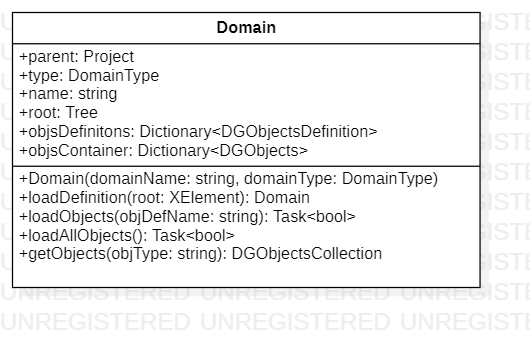
* + - 1. 类内、类外方法调用状态图

待补

* + 1. ProjectList详细设计
       1. 类图



* + 1. Domain详细设计
       1. 类图



* + - 1. 属性及方法描述

待补

* + - 1. 类核心方法流程

同图 3.3.1.3

* + - 1. 类内、类外方法调用状态图

待补

* + 1. DGObjects详细设计
       1. 类图

1. 收藏类型：由 HttpRequest 获得｛图片／专辑｝
2. 添加／取消收藏：由 HttpRequest 获得
3. 专辑 ID：由 HttpRequest 获得
4. 当前用户：由 Session 获得
   * + 1. 属性及方法描述

|  |  |  |  |
| --- | --- | --- | --- |
| class DGObjects | 变量 | Domain parent | 对象组的父节点 |
| DGObjectsDefinition definition | 对象组的定义 |
| DataSet rawDataSet | 对象组的源数据 |
| Dictionary<DataRow, DGObject> rowView2Obj | 数据行和对象索引 |
| Dictionary<DGObject, DataRow> obj2RowView | 对象和数据行索引 |
| 方法 | DGObject this[string key] | 获取 name 对应的对象 |
| DGObject this[int id] | 获取 id 对应的对象 |
| bool load() | 加载对象组 |

* + - 1. 核心方法流程

同图 3.3.1.3

3.3.5.4 类内、类外方法调用状态图

待补

* + 1. DGObject详细设计
       1. 类图

待补

* + - 1. 属性及方法描述

|  |  |  |  |
| --- | --- | --- | --- |
| class DGObject | 变量 | int id | 对象 ID |
| string name | 对象名称 |
| string fullName | 对象全称 |
| string description | 对象描述 |
| DataRow rawData | 对象对应的数据行 |
| 方法 | virtual bool LoadObjs(DGObjects objs) | 加载数据库到对象 |
| virtual List<DataView>  tableViews(IEnumerable<DGObject> objs) | 对象对应的表视图 |
| virtual List<FrameworkElement> chartViews  ( IEnumerable<DGObject> objs, double width, double height) | 对象对应的图视图 |

* + - 1. 核心方法流程

同图 3.3.1.3

* + - 1. 类内、类外方法调用状态图

待补

* + 1. 详细设计
       1. 属性及方法描述
       2. 核心方法流程
       3. 类内、类外方法调用状态图

界面类

4.1 类描述

界面类使用了C#的WPF框架，每个类对应一个页面（.xaml文件），用于实现页面的业务逻辑。核心的

MainFrame类继承了IMainFrame接口， 属于MVC的 Controller层。

4.2 功能

1. 登录；
2. 配置Server端IP及端口；
3. 管理项目列表信息；
4. 查看项目地址信息；
5. 查看项目图层信息；
6. 查看项目数据组信息；
7. 配置项目定义信息；
8. 配置项目图层信息；
9. 配置项目Domain信息；
10. 配置项目数据组信息；
11. 配置项目树；

4.3 类详细设计

4.3.1 UserLoginPage详细设计

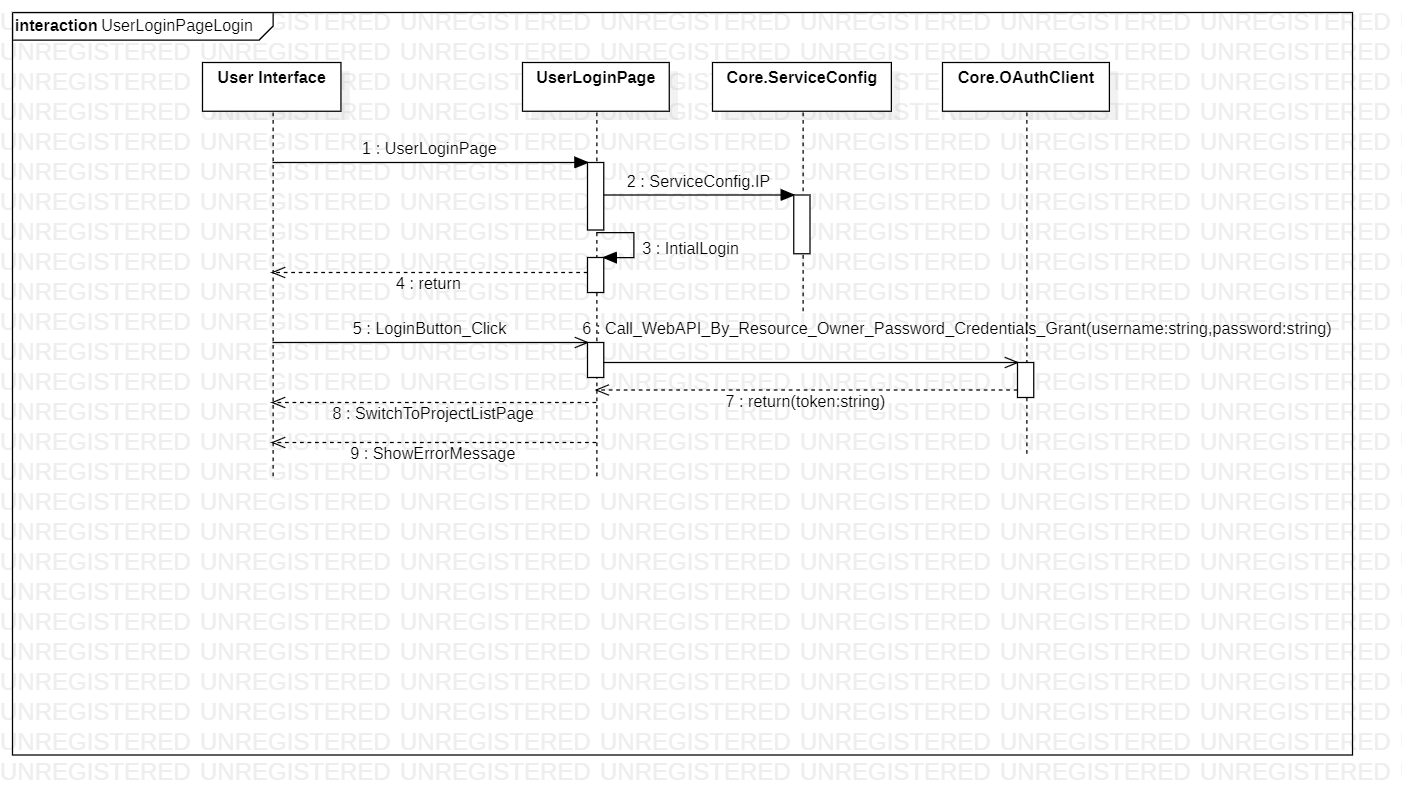
4.3.1.1 类图

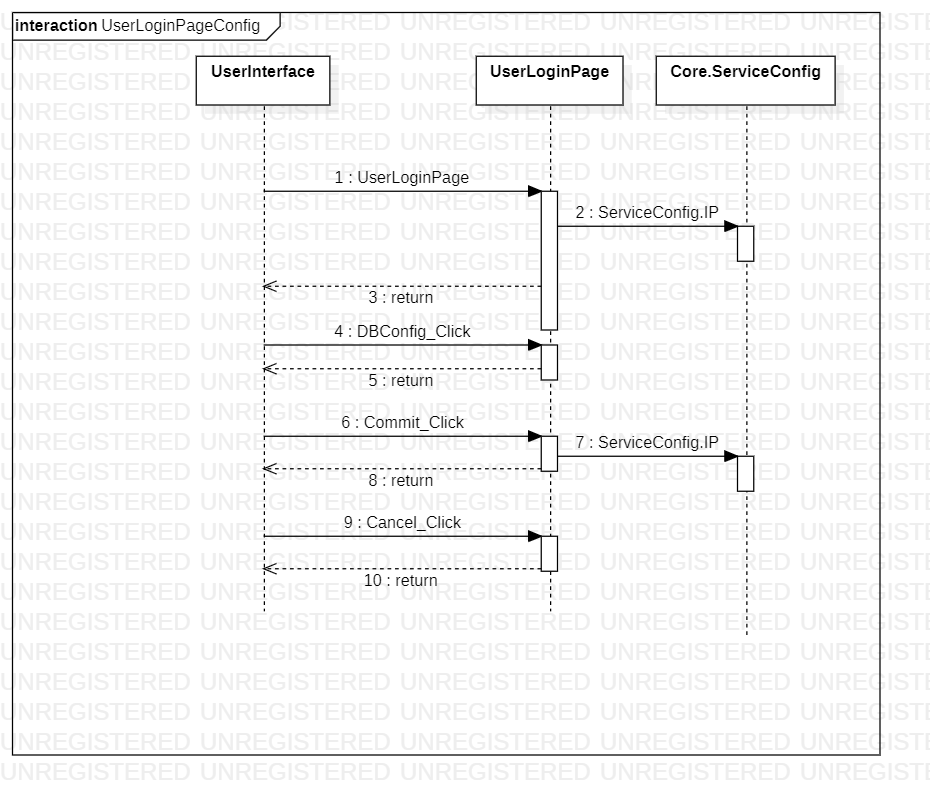


4.3.1.2 类属性及方法描述

|  |  |  |  |
| --- | --- | --- | --- |
| class UserLoginPage | 变量 | XDocument xml | 配置信息xml文件 |
| string DBAddress\_TB | Server端IP |
| string DBPort\_TB | Server端端口 |
| UserControl LoginElement | 登录组件 |
| UserControl DBConfigElement | 配置组件 |
| 方法 | LoginButton\_Click(object sender, RoutedEventArgs e) | 登录 |
| DBConfig\_Click(object sender, RoutedEventArgs e) | 切换配置组件 |
| Cancel\_Click(object sender, RoutedEventArgs e) | 取消配置 |
|  |  | Commit\_Click(object sender, RoutedEventArgs e) | 提交配置 |

4.3.1.3 类核心方法流程







4.3.2 ProjectListPage详细设计

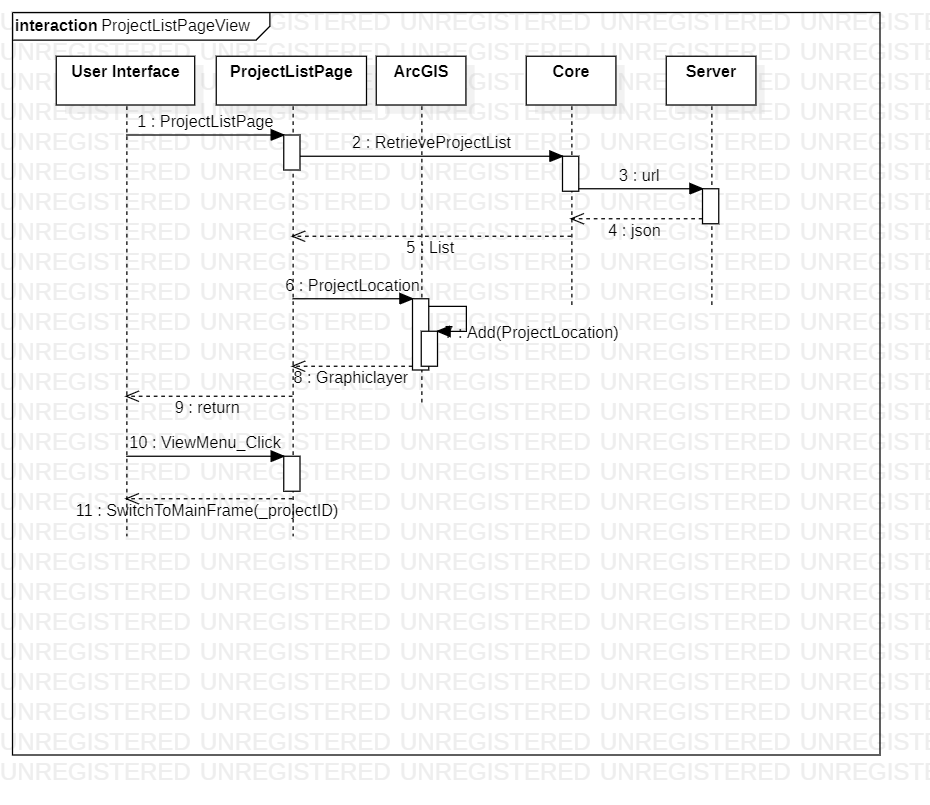
4.3.2.1 类图



4.3.2.2 类属性及方法描述

|  |  |  |  |
| --- | --- | --- | --- |
| class ProjectListPage | 变量 | ProjectList myProjectList | 项目列表 |
| string ProjectTitle | 项目名 |
| PictureMarkerSymbol \_MarkerSymbol\_Normal | 平常状态图标 |
| PictureMarkerSymbol \_MarkerSymbol\_Select | 中状态图标 |
| UserControl ProjectBox | 项目列表组件 |
| 方法 | LoadProjectListData() | 加载项目列表数据 |
| AddProjectsToMap() | 添加项目到地图上 |
| initMap() | 初始化地图事件 |
|  |  | InitializePictureMarkerSymbol() | 初始化地图图标 |

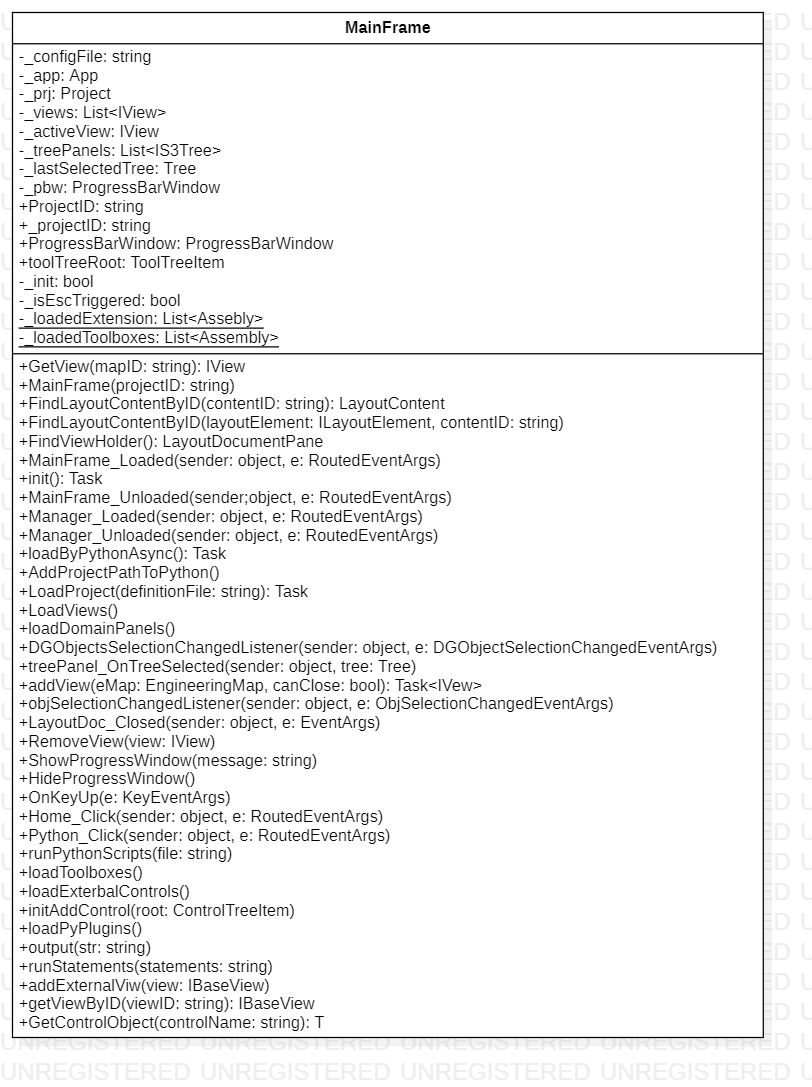
4.3.2.3 类核心方法流程





4.3.3 MainFrame详细设计

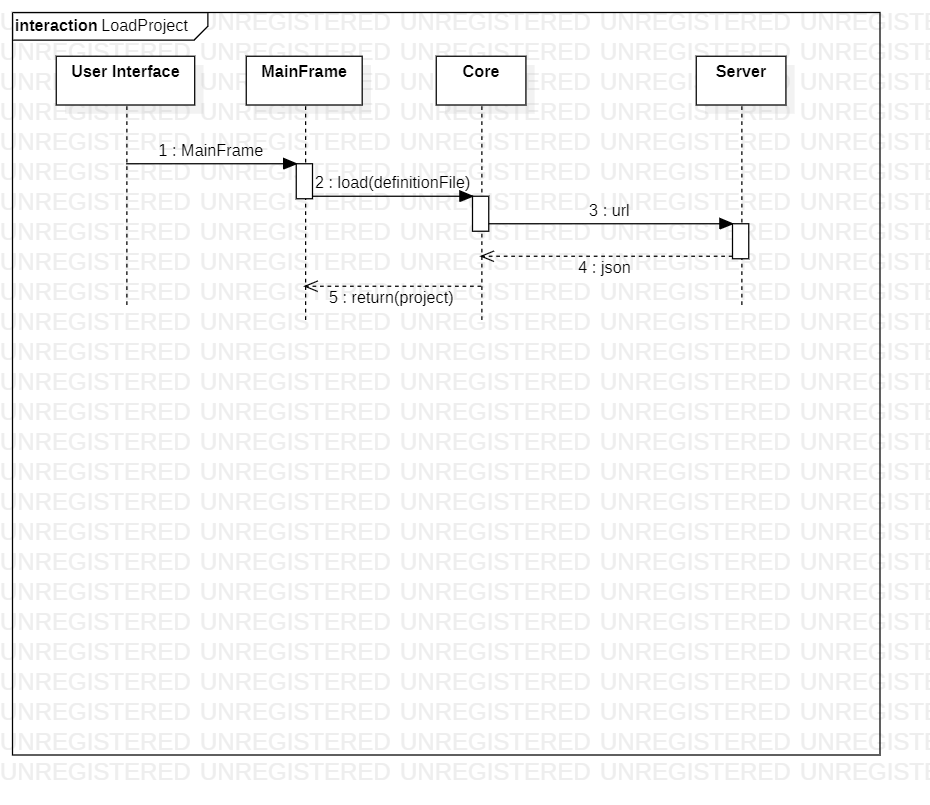
4.3.3.1 类图



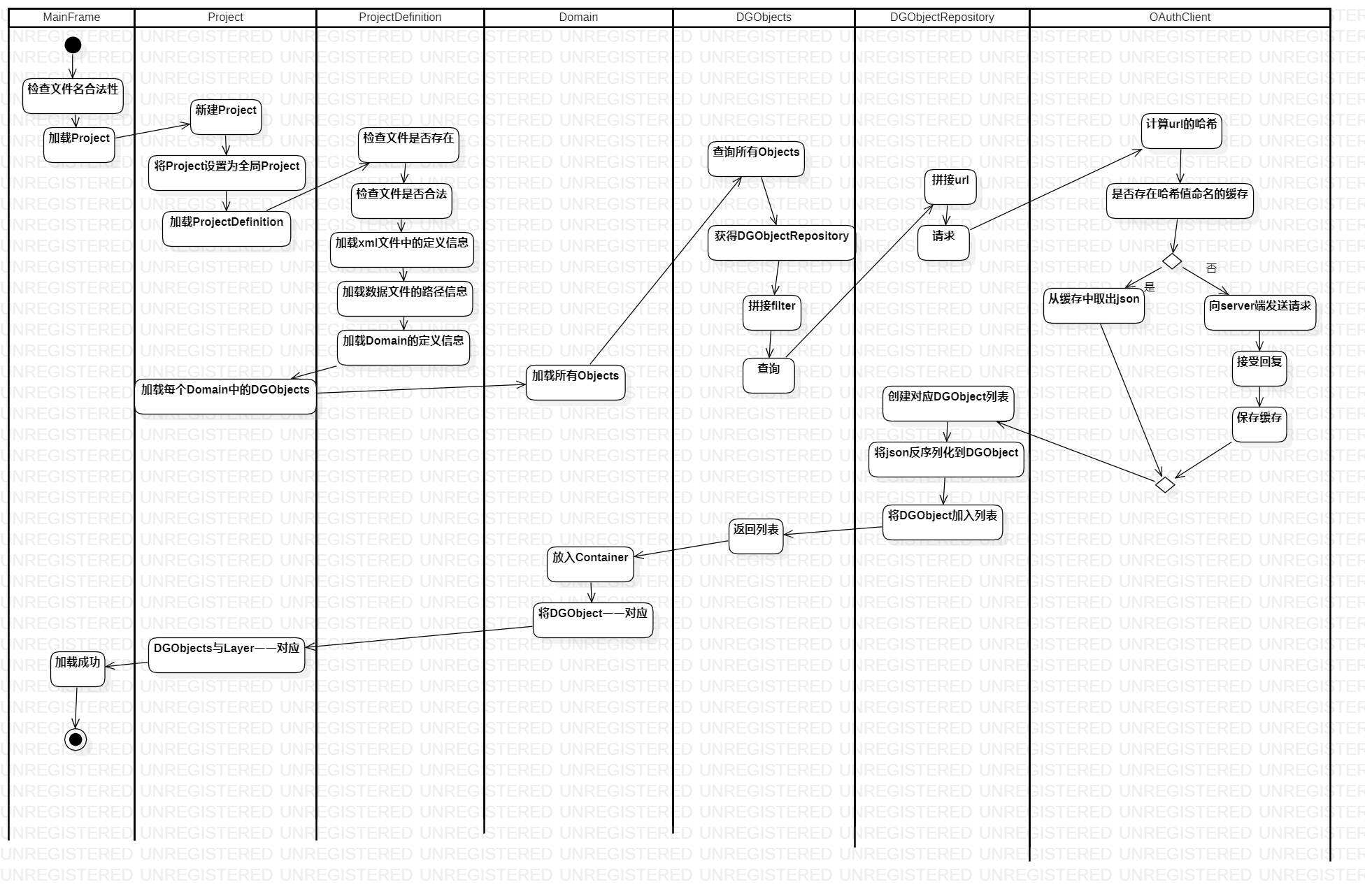
4.3.3.2 类属性及方法描述

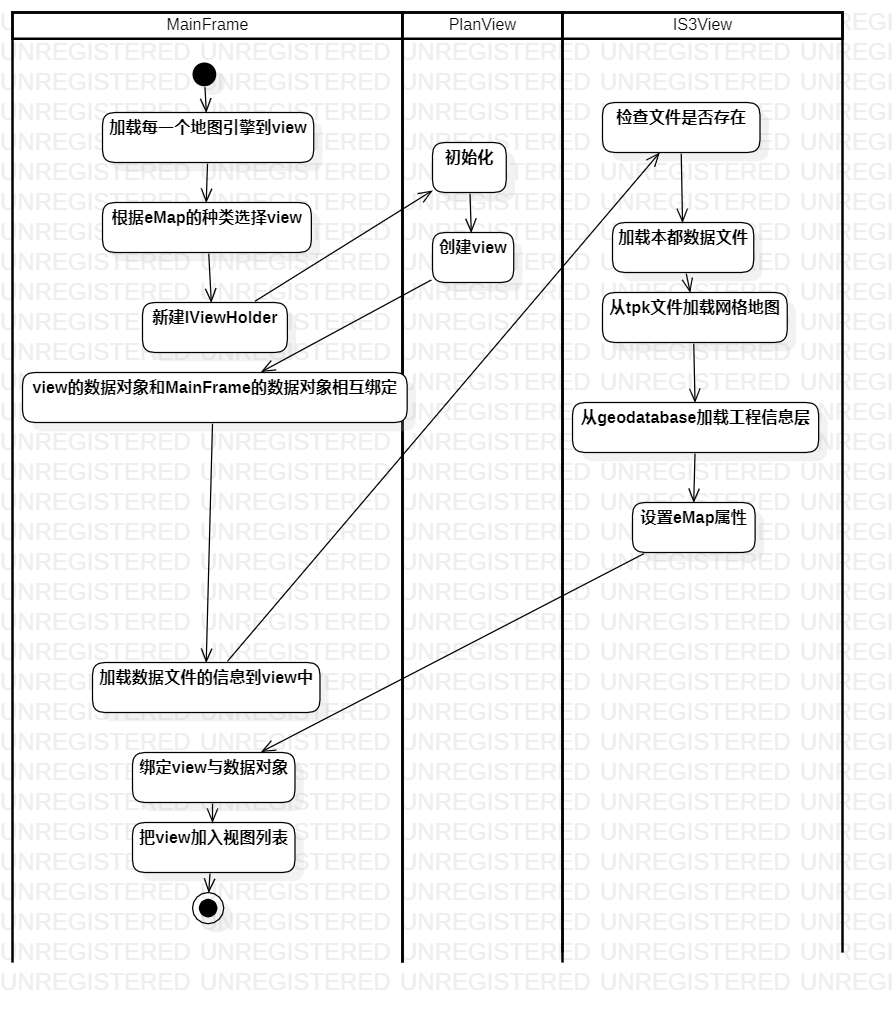
|  |  |  |  |
| --- | --- | --- | --- |
| class MainFrame | 变量 | string \_configFile | 配置信息xml文件 |
| Project \_prj | 项目 |
| UserControl PlanView | ViewHolder,视图渲染组件 |
| UserControl DataGrid | 原始数据展示组件 |
| UserControl TreePanel | 项目结构树选择组件 |
| 方法 | LoadProject(string definitionFile) | 加载项目 |
| LoadView() | 加载视图 |
| loadExtensions(); | 加载拓展库 |
|  |  | loadPyPlugins(); | 加载python插件 |

4.3.3.3 类核心方法流程



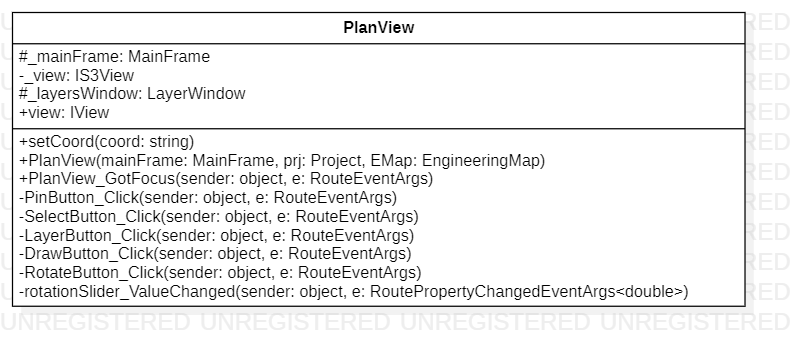


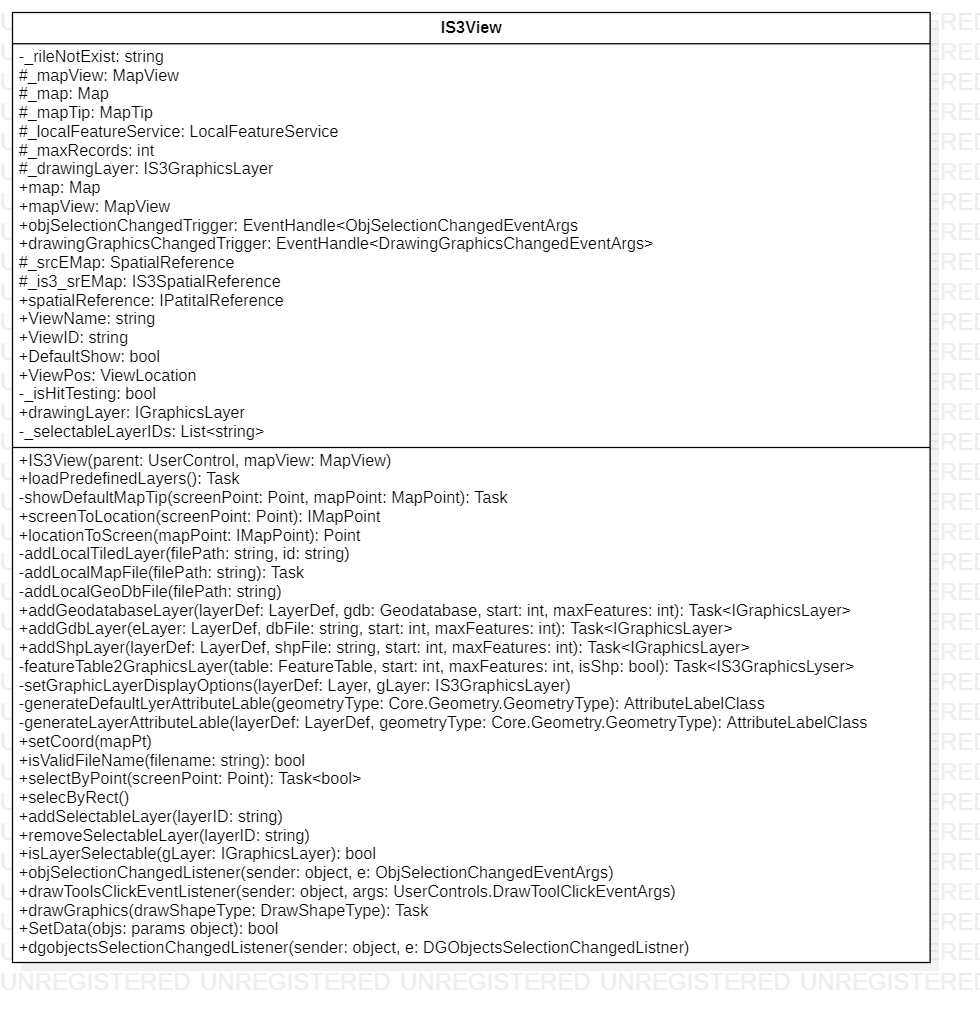




4.3.4 PlanView&IS3View详细设计

4.3.4.1 类图





4.3.4.2 类属性及方法描述

|  |  |  |  |
| --- | --- | --- | --- |
| class PlanView | 变量 | UserControl \_mainFrame | 主框架 |
| IS3View \_view | 图层 |
| 方法 | SelectButton\_Click(object sender, RoutedEventArgs e) | 选择按钮事件 |
| LayersButton\_Click(object sender, RoutedEventArgs e) | 图层按钮事件 |
| DrawButton\_Click(object sender, RoutedEventArgs e) | 绘制按钮事件 |
|  |  | RotateButton\_Click(object sender, RoutedEventArgs e) | 旋转按钮事件 |
| class IS3View | 变量 | Map map | GIS地图 |
| MapView mapView | GIS地图图层 |
| 方法 | objSelectionChangedListener(object sender,  ObjSelectionChangedEventArgs e) | 数据对象变化监听器被触发事件 |
| loadPredefinedLayers() | 加载预先设定层 |
| showDefaultMapTip(Point screenPoint, MapPoint mapPoint) | 渲染默认地图 |
|  |  | addGeodatabaseLayer(LayerDef layerDef,Geodatabase gdb, int start = 0, int maxFeatures = 0) | 添加地质数据层 |

4.3.4.3 类核心方法流程

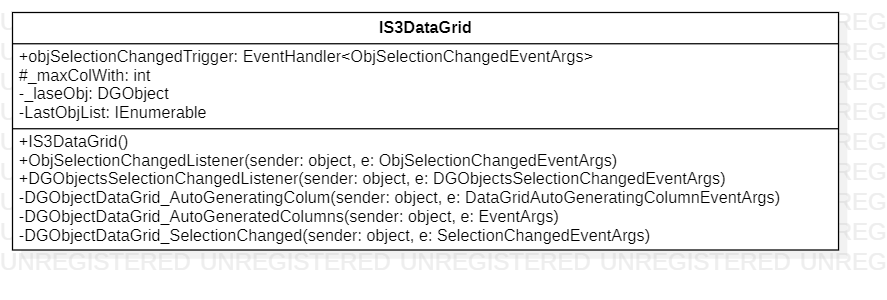
见图4.3.3.3

4.3.4.4 类内、类外调用状态图



4.3.5 IS3DataGrid详细设计

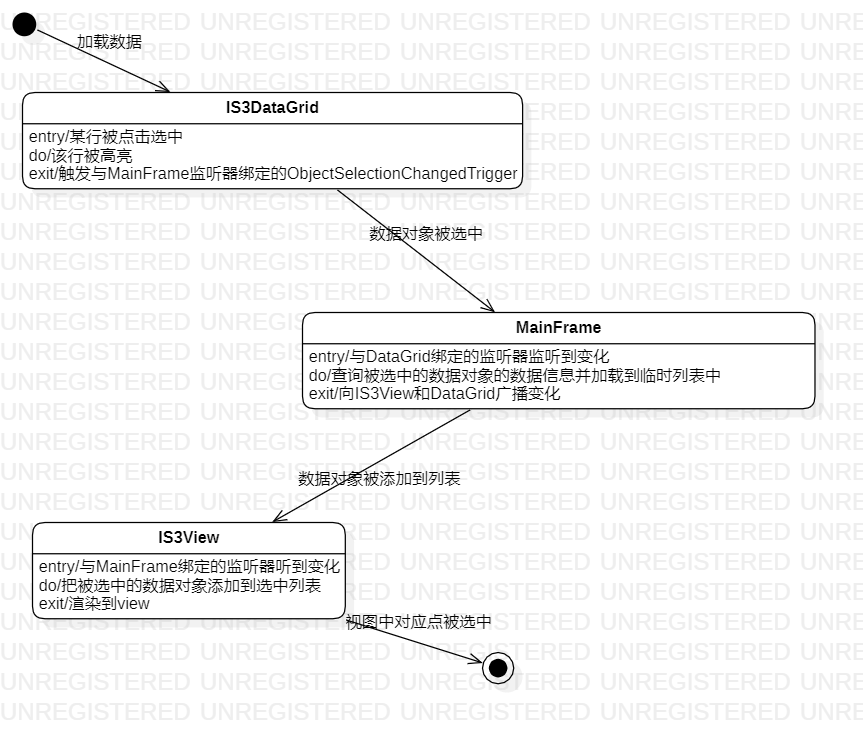
4.3.5.1 类图



4.3.5.3 类核心方法流程

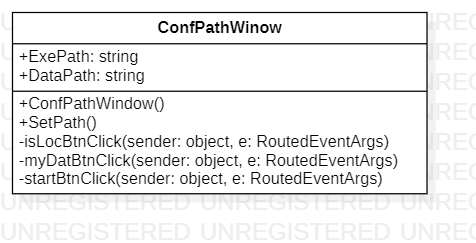
见图4.3.3.3

4.3.5.4 类内、类外调用状态图

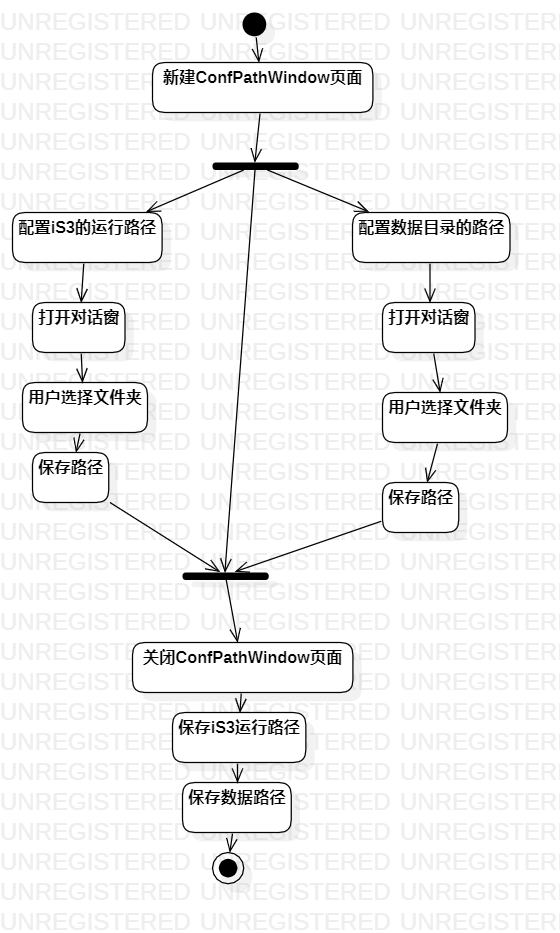


4.3.6 ConfPathWindow详细设计

4.3.6.1 类图

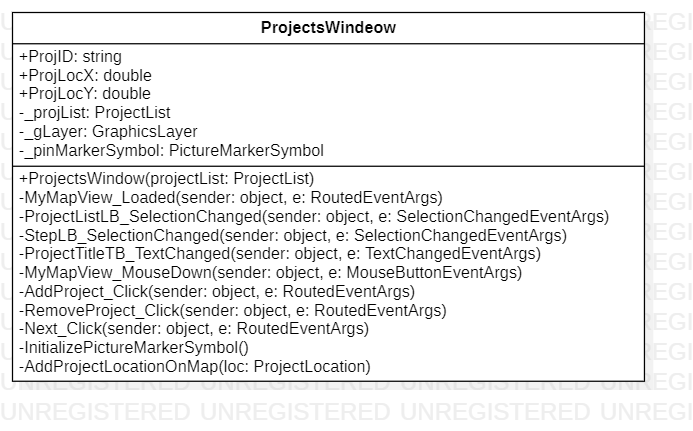


4.3.6.2 类核心方法流程



4.3.7 ProjectsWindow详细设计

4.3.7.1 类图

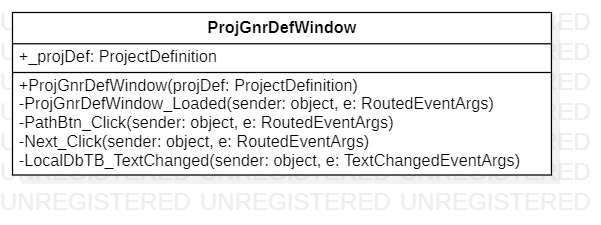


4.3.7.2 类核心方法流程

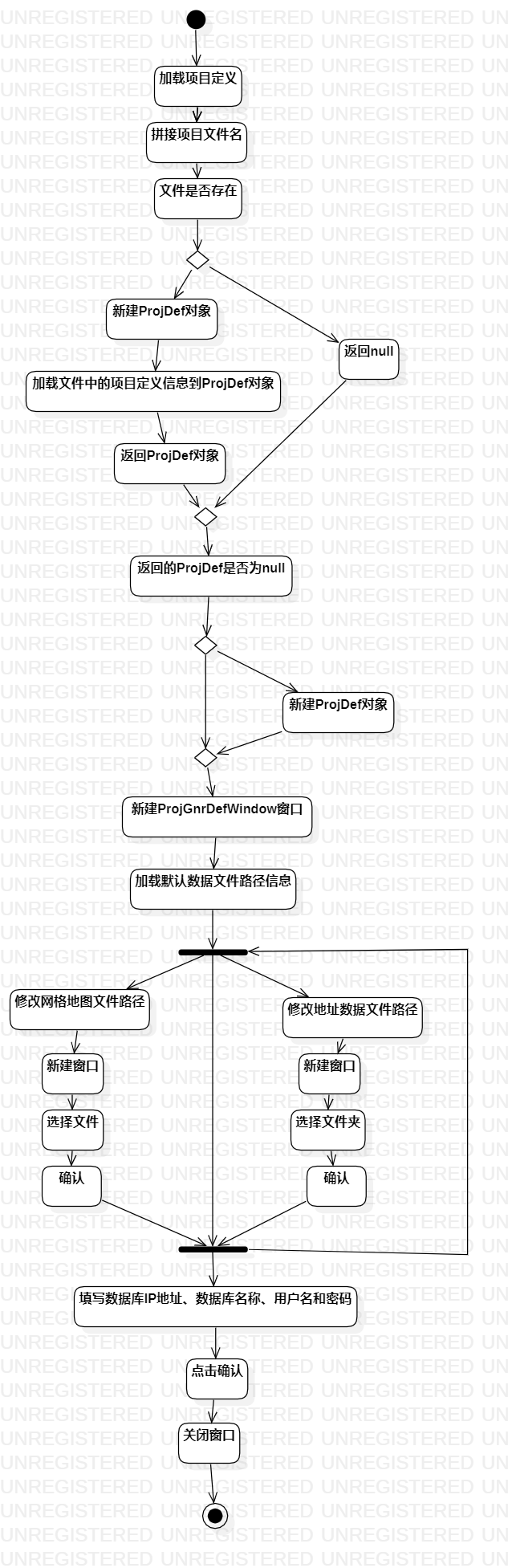


4.3.8 ProjGnrDefWindow详细设计

4.3.8.1 类图

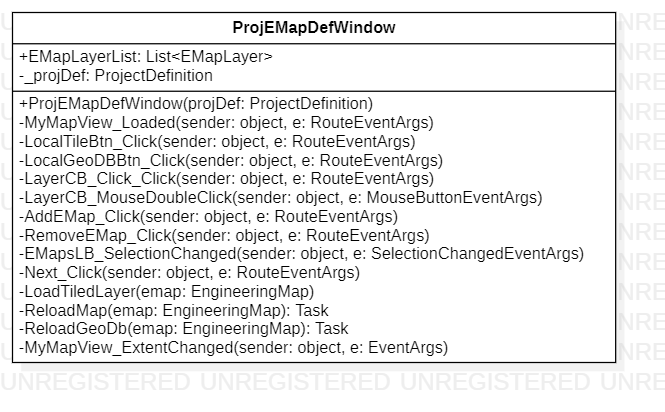


4.3.8.2 类核心方法流程

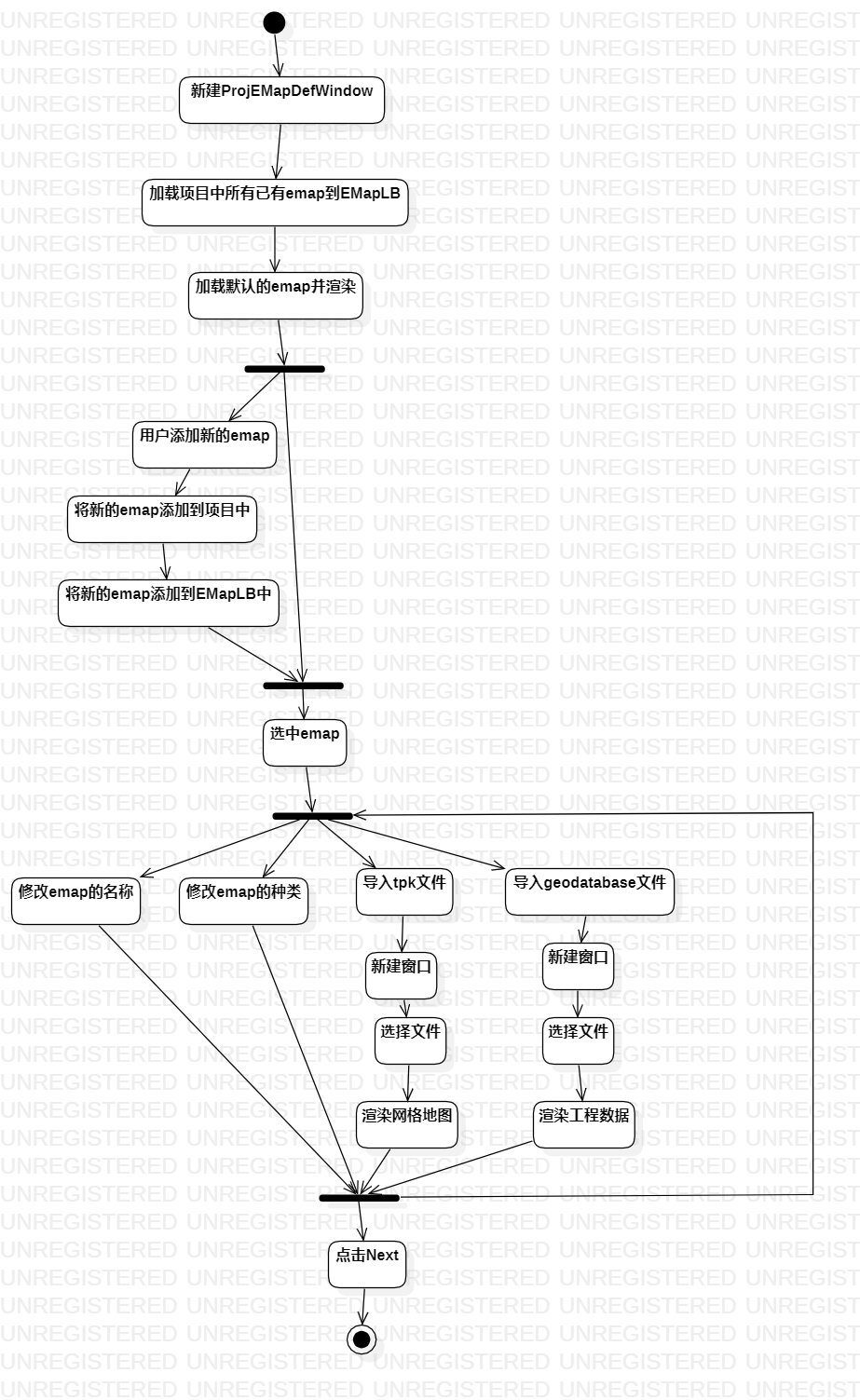


4.3.9 ProjEMapDefWindow详细设计

4.3.9.1 类图



4.3.9.2 类核心方法流程



3.10 ProjDomainDefWindow详细设计

4.3.10.1 类图

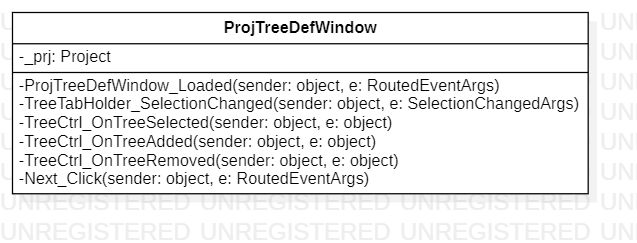


4.3.10.2 类核心方法流程



3.11 ProjTreeDefWindow详细设计

4.3.11.1 类图



4.3.11.2 类核心方法流程



控制类

5.1 描述

简称用户模块（Account Management Module）。负责用户注册、登录、登出，以及修改个人信息。

5.2 功能

1. 登录
2. 登出
3. 注册
4. 修改个人信息

5.3 模块详细设计

* + 1. 登录详细设计
       1. 输入项

1. 用户姓名：由 HttpRequest 获得
2. 用户密码：由 HttpRequest 获得
   * + 1. 输出项
3. 用户名不存在。
4. 用户密码错误。
5. 用户成功登录。
   * + 1. 逻辑流程
       2. 接口

|  |  |  |  |
| --- | --- | --- | --- |
| 请求地址： | /Login | | |
| 请求方式： | Post | | |
| 请求参数： | 字段 | 类型 | 描述 |
| name | String | 用户姓名 |
| password | String | 登录密码 |
| 成功请求响应： |  | | |
| 失败请求响应： |  | | |

* + - 1. 限制条件及出错处理

（1）用户必须已注册，若未注册，提示用户注册。

1. 用户最多输入密码次数限制，超过限制次数，禁止在 1 天内登录。
2. 数据库无法正常连接，页面显示“数据库无法正常连接”。
   * 1. 注册详细设计
        1. 输入项
           1. 用户姓名：由 HttpRequest 获得（必填）
           2. 用户密码：由 HttpRequest 获得（必填）
           3. 用户头像：由 HttpRequest 获得
           4. 用户邮箱：由 HttpRequest 获得
3. 用户城市：由 HttpRequest 获得
4. 用户性别：由 HttpRequest 获得
5. 用户生日：由 HttpRequest 获得
6. 个人描述：由 HttpRequest 获得
   * + 1. 输出项
7. 用户注册成功。
8. 用户注册失败。
   * + 1. 逻辑流程
       2. 接口

|  |  |  |  |
| --- | --- | --- | --- |
| 请求地址： | /Register | | |
| 请求方式： | Post | | |
| 请求参数： | 字段 | 类型 | 描述 |
| name | String | 用户姓名 |
| password | String | 注册密码 |
| avatarPath | String | 用户头像 |
| email | String | 用户邮箱地址 |
| city | String | 用户所在城市 |
| gender | String | 用户性别 |
| birthday | Date | 用户生日 |
| description | String | 用户个人描述 |

|  |  |
| --- | --- |
| 成功请求响应： |  |
| 失败请求响应： |  |

* + - 1. 限制条件及出错处理

（1）用户名不能重复，若已被注册，页面提示用户重新输入用户名。

1. 用户名不能超过 39 个字符，若超过，则无法继续输入。
2. 注册密码字符串可包含数字、字母、下划线，出现非法字符串，提示用户重新输 入。
3. 用户头像格式必须正确，若格式错误，提示“文件格式错误”。

（4）用户头像大小必须限制在 4M 之内，若超过最大限制，提示“图片过大”。

1. 用户邮箱地址格式必须正确。
2. 用户个人描述必须限制在 254 字之内，若超出，则无法继续输入。
   * 1. 登出接口详细设计
        1. 输入项

（1）用户 ID：由 Session 获得

* + - 1. 输出项

1. 用户成功注销。
2. 用户注销失败。
   * + 1. 逻辑流程
       2. 接口

|  |  |  |  |
| --- | --- | --- | --- |
| 请求地址： | /Logout | | |
| 请求方式： | Get | | |
| 请求参数： | 字段 | 类型 | 描述 |

|  |  |
| --- | --- |
| 成功请求响应： |  |
| 失败请求响应： |  |

* + - 1. 限制条件及出错处理

1. 用户必须已登录。
2. 数据库无法正常连接，页面提示“数据库无法正常连接”。
   * 1. 修改个人信息详细设计
        1. 输入项
3. 用户姓名：由 HttpRequest 获得
4. 用户旧密码：由 HttpRequest 获得
5. 用户新密码：由 HttpRequest 获得
6. 用户确认密码：由 HttpRequest 获得
7. 用户所在城市：由 HttpRequest 获得
8. 用户个人描述：由 HttpRequest 获得
   * + 1. 输出项

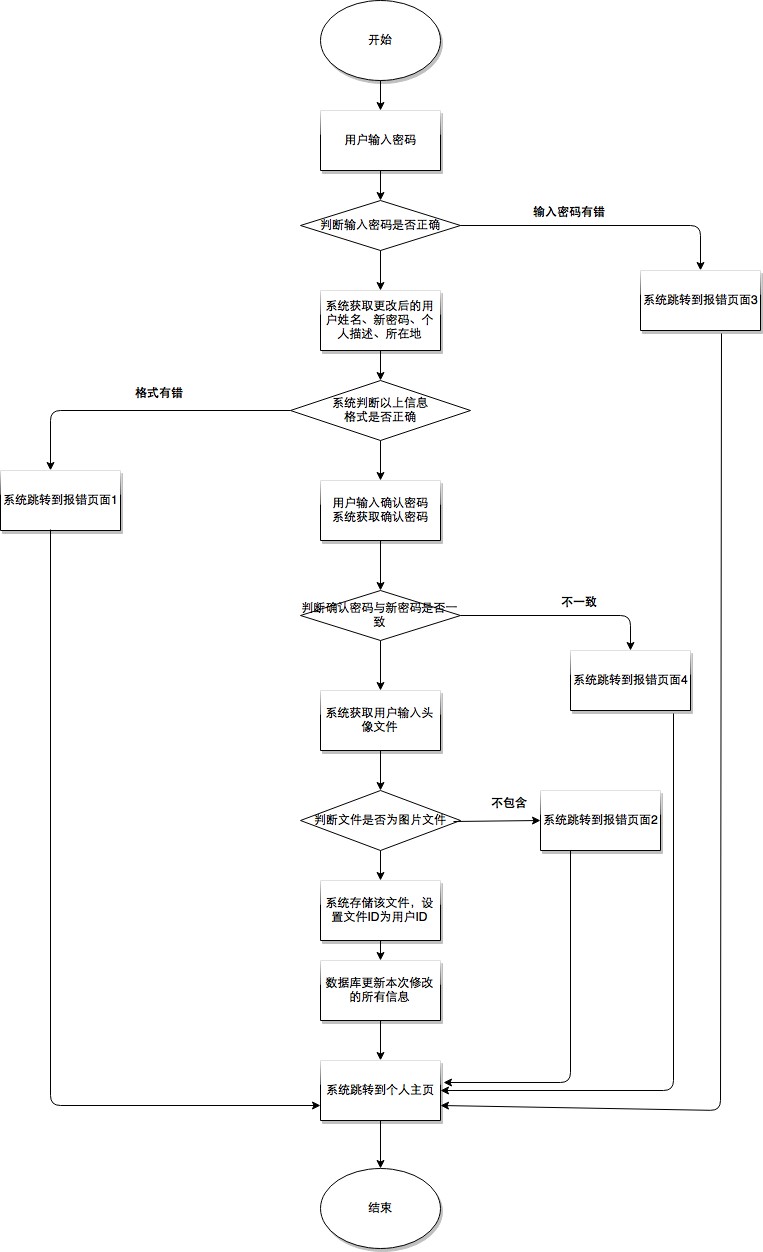
完成修改个人信息

* + - 1. 接口

|  |  |  |  |
| --- | --- | --- | --- |
| 请求地址： | /ModifyPersonal | | |
| 请求方式： | Post | | |
| 请求参数： | 字段 | 类型 | 描述 |
| name | String | 用户姓名 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | opassword | String | 用户旧密码 |
| npassword | String | 用户新密码 |
| cnpassword | String | 用户确认密码 |
| gender | String | 用户性别 |
| city | String | 用户所在城市 |
| description | String | 用户个人描述 |
| inputFile | Part | 用户输入文件 |
| 成功请求响应： |  | | |
| 失败请求响应： |  | | |

* + - 1. 逻辑流程



# 图 5.3.4.4——修改个人信息流程图

* + - 1. 限制条件及出错处理

1. 用户名不能重复，若已被注册，页面提示用户重新输入用户名。
2. 用户名不能超过 39 个字符，若超过，则无法继续输入。
3. 注册密码字符串可包含数字、字母、下划线，出现非法字符串，提示用户重新输 入。
4. 用户头像格式必须正确，若格式错误，提示“文件格式错误”。

（4）用户头像大小必须限制在 4M 之内，若超过最大限制，提示“图片过大”。

1. 用户邮箱地址格式必须正确。
2. 用户个人描述必须限制在 254 字之内，若超出，则无法继续输入。
3. 用户必须正确输入旧密码，若错误，提示重新输入。
4. 用户确认密码必须和新密码相同，若不相同，提示用户重新输入确认密码。