

# PKU To-gather 程序功能介绍

- PKU To-gather为提供了一个集组织策划活动、宣传征集参与者、报名参与活动、社交交友等功能于一体的软件平台。
- 功能：
  1. 注册账号；登录
  2. 设置头像、昵称、个性签名和活动偏好等个人信息
  3. 在活动总览界面进行活动的组织发布或加入其他人发布的活动，每个活动都有活动类型选项、活动细节介绍等属
  4. 好友系统，支持用户进行好友查询、好友信息（比如活动偏好、个性签名等）浏览、好友添加。



# 全局管理类 (overall)

全局（静态）类 overall 来管理数据和核心功能。做到类之间的数据传递。

## 成员变量：

QSettings\* settings: 持久化存储用户信息（包括昵称头像、好友列表等，以用户的id作为索引）。

person\* me: 指向当前登录用户的 person 对象。

QVector<activitys\*> allActs: 存储所有活动对象的指针。

MainWindow\* activityList: 指向主窗口的指针，方便其他模块（如 publishWindow）直接更新主窗口的活动列表。

## 成员函数：

saveActivitiesToFile(): 将 overall::allActs 中的所有活动数据（名称、详情、发布者、参与者列表等）序列化为 JSON 格式，并保存到本地文件（saveActivitys/activities.json）。

loadActivitiesFromFile(): 在程序启动时，从 JSON 文件中读取活动数据，并填充到overall::allActs。

# 程序入口 (main.cpp)

## 功能：

初始化 QApplication。使用 qRegisterMetaType 注册自定义类型 (person\*, activitys\*)，以便它们能被存储在 QVariant 中，从而能在 QListWidget 的 setData 中使用。

初始化 overall 全局对象，如 settings 和 MainWindow。在主窗口显示前调用 overall::loadActivitiesFromFile()，加载所有持久化的活动数据。创建并显示初始界面。

## 活动 (activities)

### **activitysData类:**

活动的各方面信息，包括

activityName, activityDetails, activityTag[5]（活动的分类）， posterName, players(所有报名参加该活动用户的昵称)。

### **activitys类:**

设置和获取活动信息（如名称、详情、标签）；

管理参与者列表，如添加参与者（addPlayer）、获取参与者数量（playerNum）和获取特定参与者姓名（playerIName）。

通过写入json进行数据的保存。

## 用户类 (user, person)

### **person类**

代表一个用户。包含用户的基本信息，如 userName、password、nickname 等。

好友关系通过 QStringList friends 存储。可以添加和删除好友。

用户信息利用 QSettings 进行持久化。

### **User类**

loadUserData(): 从 overall::settings 中加载当前用户的昵称、签名、头像路径等信息并显示在界面上。

on\_pushButtonSignature\_clicked() 等实现各项用户信息的修改。

## 主界面(mainmenu)

先要求用户进行注册或登录 (login) ; 然后, 进入活动界面(mainwindow)、用户信息界面(user)或交友界面(social)

## 活动界面 (MainWindow)

用于总览所有活动。以列表形式展示 overall::allActs 中的所有活动。

通过 item->setData(Qt::UserRole, ...) 将 activitys 指针与列表项关联。提供发布、查看和删除活动的入口。

发布活动: 进入publishWindow

查看详情: 双击列表中的任何活动项, 会创建一个 actDetailWindow 对话框来显示该活动的详细信息。

删除活动: 右键点击活动项会弹出上下文菜单。如果选择删除, 程序会检查当前用户是否为活动发布者。只有发布者才有权限删除, 删除后会同步更新 overall::allActs 和持久化存储。

## 活动详情窗口 (actDetailWindow)

**功能：**用于展示单个活动的详细信息并允许用户报名。接收一个 `activitys*` 指针，并将其详细信息（标题、详情、标签、已报名用户列表）显示在界面上。

**报名：**用户点击“报名”按钮 (`on_signBntm_clicked`) 后，则（如果未报名）将当前用户的昵称 (`overall::me->nickname`) 添加到该活动的 `players` 列表中。调用 `overall::saveActivitiesToFile()` 将更新后的参与者列表保存到文件。在界面上实时更新参与者列表。

## 发布活动窗口 (publishWindow)

**功能：**用于创建新活动。让用户输入新活动的名称、详情，并选择标签。

点击“发布”按钮 (`on_publishButton_accepted`) 后：创建一个新的 `activitys` 对象。用界面上输入的数据填充该对象，并将发布者设置为当前用户 (`overall::me->userName`)。将新创建的 `activitys` 指针添加到 `overall::allActs`。在 `MainWindow` 的 `QListWidget` 中添加新活动的可视化条目。调用 `overall::saveActivitiesToFile()` 持久化新活动。

## 好友详情窗口 (friendDetailDialog)

展示某个好友的个人信息。

接收一个好友的 username。根据这个 username 从 overall::settings 中读取并展示该好友的昵称、签名、头像和兴趣爱好。

删除好友：点击“删除”按钮后，程序会：从当前用户的 settings 中找到“friends”列表，并移除该好友的 username。还会尝试从对方的“friends”列表中移除当前用户，实现双向关系解除。

## 分工：

- 徐畅：负责所有活动相关的类（前端及后端）
- 何恩赐：负责所有ui设计，包括qt直接图形设计和代码方式
- 解成毅：负责所有用户相关的类（前端及后端）

# 总结与反思

- 1. 项目具有较高使用价值，并且我们的程序已经能够满足一些生活需要。
- 2. 基本实现了所有预期效果。其中的一些难点比如：（1）ui设计的美观性；（2）数据存储；（3）活动显示出参与人，“加好友”操作等也都得到了克服。
- 3. 通过github共享文件及实时沟通，高效地完成了程序。

## 反思：

- I. 项目虽然工作量不小，但在难度上欠缺（比如，完全没有使用到任何算法）
- II. 对面向对象程序设计的使用的依然稚嫩，没有使用到继承、多态（当然，这可以归因于程序实现的功能有限），代码的复用性有待提升