

# 《卡路里战友》App

## 1. 项目简介 (Project Brief)

- **项目名称:** 卡路里战友 (Calorie Buddies)
- **核心目标:** 帮助大学生在繁重的学业压力下，通过记录饮食、设定目标和社交激励，来管理自己的卡路里摄入和体重。
- **技术架构:** JavaFX (图形客户端) + Java Socket (多线程服务端)
- **团队人数:** 3 人
- **项目周期:** 6 周

## 2. 核心功能规划

### A. 核心功能 (本地实现)

1. **用户档案 (User Profile):**
  - 在本地创建和存储用户数据 (无需注册登录)。
  - 输入和编辑个人信息：身高、当前体重、年龄、性别、日常活动水平 (如：久坐、轻度运动等)。
2. **目标与推荐 (Goal & Recommendation):**
  - **核心算法:** 根据用户档案信息，自动计算用户的 **BMR** (基础代谢率) 和 **TDEE** (每日总能量消耗)。
  - **目标设定:** 用户可以设定目标 (例如：6周内减重 2kg，或保持体重)。
  - **智能推荐:** App 根据目标，自动计算出“每日推荐卡路里摄入量”(例如：TDEE - 500 大卡)。
3. **饮食日志 (Food Log):**
  - **本地食物库:** 内置一个简单的食物数据库 (可以用 JSON 或 CSV 文件存储，包含食物名称和卡路里)。
  - **记录功能:** 用户可以从数据库搜索食物，并记录摄入的份量 (如：米饭 200g)。
  - **自定义添加:** 允许用户手动输入“自定义食物”及其卡路里。
  - **主页看板:** 首页清晰显示“目标摄入”、“已摄入”、“剩余额度” (例如用进度条显示)。
4. **打卡与统计 (Check-in & Stats):**
  - **一键打卡:** 每天完成记录后可以点击“打卡”。
  - **统计图表:** 显示连续打卡天数 (激励!)。
  - **历史回顾:** 简单的日历视图，可以查看过去几天的卡路里摄入历史。

### B. 进阶功能 (社交网络)

这部分是项目的亮点，也是网络编程的重点。

1. **服务端 (Server):**
  - 使用 Java `ServerSocket` 搭建一个多线程服务器。
  - 服务器负责处理用户注册、登录、数据存储和“朋友圈”消息转发。
  - **数据存储:** 为简化开发，初期可使用文本文件 (如 JSON 或序列化对象) 在服务器端存储用户信息和帖子，如果时间充裕再考虑使用数据库 (如 **SQLite**)。
2. **客户端网络功能 (Client):**

- **注册与登录:** 启动 App 时，连接到服务器进行注册或登录。
- **“公共广场”(朋友圈简化版):**
  - **为什么简化?** 6周内实现一个完整的“好友系统”(加好友、同意/拒绝)非常耗时。
  - **建议方案:** 做一个“公共广场”(Public Feed)。所有人发的帖子，所有人都能看到。这同样满足了社交分享和激励的需求，但开发量减半。
- **分享功能:** 用户可以将自己的“今日饮食总结”或“打卡成功”一键分享到“公共广场”。