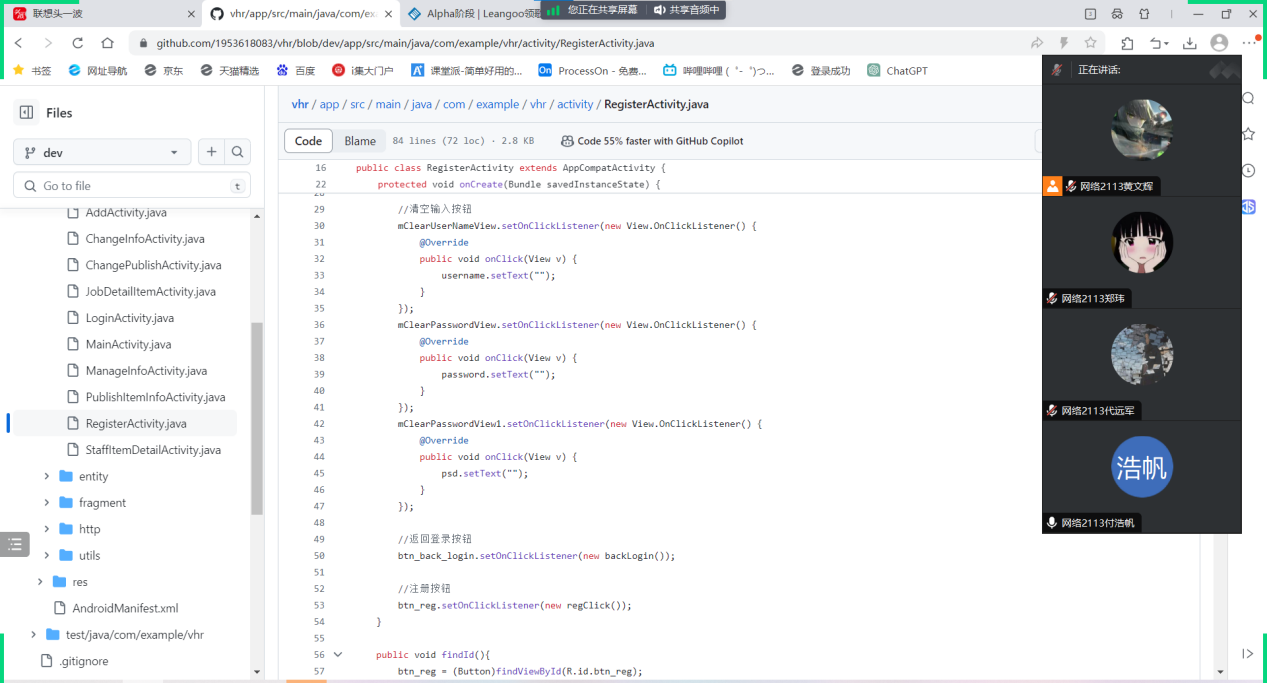
### 事后诸葛亮分析——alpha阶段

1. 事后分析——复盘



## 设想和目标

1. **我们的软件要解决什么问题？是否定义得很清楚？是否对典型用户和典型场景有清晰的描述？**

本次我们的团队需要完成的是一个微人事系统旨在解决企业人力资源管理的效率问题。它通过数字化和自动化流程，简化和优化HR日常工作，包括招聘、员工管理、薪酬计算、绩效考核等方面。微人事系统的Android端扩展有明确的目标和功能范围：包括但不限于薪资查询、员工管理、招聘信息发布等，不同角色有不同的功能权限。典型用户：

企业员工：希望能够方便快捷地进行日常人事相关操作，如请假、打卡、查看薪资等。

部门经理：需要审批员工的请假和出差申请，查看团队的考勤和绩效数据。

HR人员：需要管理全公司的员工信息，处理招聘、入职、离职等流程，统计和分析人事数据。

典型场景：

外勤人员打卡：员工在外出差或现场工作时，通过Android设备定位功能进行考勤打卡。

请假和出差申请：员工随时随地提交请假或出差申请，经理可以实时审批，HR可以即时记录。

薪资查询和绩效反馈：员工可以通过移动端查看自己的薪资明细和绩效考核结果，并进行反馈和沟通。

通知和公告：HR发布的通知和公司公告可以通过移动端即时推送给所有相关人员。

1. **我们达到目标了么（原计划的功能做到了几个？按照原计划交付时间交付了么？ 原计划达到的用户数量达到了么?）**

达到目标了，原计划的功能有：登录、注册、信息验证、数据获取、数据修改、筛选信息、查看信息、查看信息分析。没有按照原计划时间交付，到了原计划时间尚有功能未完善。还没有达到更多的用户数量，因为现在app还不够完善。

1. **和上一个阶段相比，团队软件工程的质量提高了么？ 在什么地方有提高，具体提高了多少，如何衡量的？**

提高了，主要是思想上或者说是感性上学到了很多东西。开始的时候，大家对项目的未来都抱有美好的想象，但随着时间的推移，技术上的困难和课程上的压力都让大家的热情冷却，但是我们没有放弃，我们努力突破自己，突破的不仅仅熬夜，突破是自己的心理承受能力。一次次的压力让我们不断学会承受，让我们强大。

1. **用户量, 用户对重要功能的接受程度和我们事先的预想一致么? 我们离目标更近了么?**

用户对功能的接收程度与我们事先预想的存在较小的差别，主要在于不同人群用户使用我们项目的目的与我们的预期不同，呈现混合情况，即不同人群询问内容差别不大。为了得出真实的使用情况，我们会积极分享给身边的朋友。

1. **有什么经验教训? 如果历史重来一遍, 我们会做什么改进?**

如果重来一遍，我们会在直接自己搭建服务器端或者扩展本地数据库以确保扩展功能更加的丰富。

**计划：**  
**1. 你原计划的工作是否最后都做完了？如果有没做完的，为什么？**  
 原计划的工作大部分已经完成，可能有一些细节问题没有实现，为了进一步增加用户体验的好感度，还需要在第二阶段进一步完善。原因之一是我们初期计划未考虑到这些细节，导致没有进一步完善。

**2. 有没有发现你做了一些事后看来没必要或没多大价值的事？**  
 很少，一般都是组长安排给我们任务，我们各人完成自己相应的任务。  
  
**3. 是否每一项任务都有清楚定义和衡量的交付件？**  
 我们队每个人每周的工作量都有清晰的定义，每个人的任务都分配的很清楚。  
**4. 是否项目的整个过程都按照计划进行，项目出了什么意外？有什么风险是当时没有估计到的，为什么没有估计到？**  
 项目的绝大部分过程都是按照计划进行，出现的意外为服务器源码不在我方，容易出现服务器卡死的风险，这个风险就是没有估计到的，因为服务器是上界学长搭建的，我们只使用接口进行数据交互，服务器端出现问题我们是完全无法调整的。  
**5. 在计划中有没有留下缓冲区，缓冲区有作用么？**  
 有留下缓冲区，因为在开发过程中总是会遇到种种问题的，例如遇到了瓶颈，周一至周五需要上课，项目推进受到一定的影响。所以为了按时完成开发，缓冲区还是很有必要的  
**6. 将来的计划会做什么修改？（例如：缓冲区的定义，加班）**  
 将来的计划会在任务的时间设定上做修改，因为已经进入了考试月，所以需要设置更多的换冲区，但是如果越过缓冲区还是没完成任务，就需要在深夜进行加班对与需求分析的理解组员们需要一致，分工之后，组员间还是需要加强沟通。

**资源：**

1. **我们有足够的资源来完成各项任务吗？**

答：有，我们团队有六名成员，关于整体项目的分工合理，专门做了一个问卷来收集大家对招app的看法和必要功能需求。  
**2. 各项任务所需的时间和其他资源是如何估计的，精度如何？**

答：所需时间都按照任务的难度进行设置，我们是按照任务平均分配给每天的任务量，偶尔会出现当天的任务较重，没有及时完成任务，顺延到后一天的情况。总体来说还是有按照计划来实现任务。  
**3. 测试的时间，人力和软件/硬件资源是否足够？对于那些不需要编程的资源，是否低估难度？**

答：测试时间和人力软件、硬件资源还是足够的。但是对于美工和文案方面对我们都是理工科的学生来说是有点低估了难度，还是需要去网上参考借鉴大家的设计和文案撰写才能有更好的思路来完成我们的项目。  
**4. 你有没有感到你做的事情可以让别人来做（更有效率）？**

答：暂时没有，我们团队人员配置是比较精简的，分工也是按照每个人的特长进行分配任务的，所以每个角色都有很好的完成任务。  
**5. 有什么经验教训？如果历史重来一遍，我们会做什么改进？**

答：每个人都做一份自己的每日计划表，把各自的时间都协调好，尽量不要和其他课程的任务冲突。

**变更管理**

1. **每个相关的员工都及时知道了变更的消息?**

确保每个相关的团队成员都及时知道变更消息，可以采取以下措施：  
 通过内部通知、团队会议、即时消息等方式通知成员（微信、QQ、电话等）；  
 利用项目管理工具发送邮件、更新任务或创建讨论来通知成员；  
 定期召开培训会议，让成员了解新流程和规定。  
**2. 我们采用了什么办法决定“推迟”和“必须实现”的功能?**  
 可以考虑使用变更管理流程来评估推迟和必须实现的功能。例如，使用成本预算、时间表和资源可用性等指标进行权衡，以决定哪些功能必须在当前版本中实现，哪些可以推迟到以后的版本中添加。团队应该在评估过程中考虑安全性、紧急程度和影响范围等因素。  
**3. 项目的出口条件（Exit Criteria）是否得到清晰的定义?**  
 在软件工程项目中，出口条件可以定义为达成某一里程碑或完成某一功能所需满足的标准。例如，在开发一个软件应用程序时，出口条件可以包括实现必要的功能、通过所有测试用例、完成用户界面设计等。确定出口条件可以帮助团队跟踪进度，了解项目的状态，确保项目能够按时完成。  
**4. 对于可能的变更是否能制定应急计划?**  
 团队需要制定应急计划来快速响应可能的变更。例如，可以建立一个流程来评估和批准任何变更，并创建备用方案以便在必要时应对不可预见的问题。团队应该定期审查和更新应急计划，确保它们能够适应变化和降低风险。  
**5. 员工是否能够有效地处理意料之外的工作请求？**  
 成员需要通过各种培训和学习机会来学习如何有效处理意料之外的工作请求。例如，可以为成员提供关于特定工具、技术和最佳实践的培训，以帮助他们更好地处理工作请求。此外，建议团队建立相互合作的文化，使成员可以共享和学习他们的工作方法，并及时向同事或上级寻求帮助。例如，如果一个成员需要在短时间内完成一项任务，而他没有经验或知识来完成这项任务，他可以向更有经验的同学寻求帮助，或向老师寻求支持。

**设计/实现**

1. **设计工作在什么时候，由谁来完成的？是合适的时间，合适的人么？**

前端设计工作主要是由美工人员负责的，在项目初期进行。在每个功能具体实现之前先由美工/前端负责人员对该功能的前端页面展示进行设计，之后后端负责人在该前端页面完成之后进行后端代码编写，同时前端人员能够对前端页面开展进一步的美化工作。在功能研发期间，前后端及美工各司其职，我们认为是较为合适的时间和人选。

1. **设计工作有没有碰到模棱两可的情况，团队是如何解决的？**

由于我们初步的需求分析与功能设计已经对需要实现的功能做出了较为明确的规定，因此模棱两可的情况还是较少出现的。

1. **团队是否运用单元测试（unit test），测试驱动的开发（TDD）、UML, 或者其他工具来帮助设计和实现？这些工具有效么？ 比较项目开始的 UML 文档和现在的状态有什么区别？这些区别如何产生的？是否要更新 UML 文档？**

我们团队主要使用GitHub，Leangoo等软件来帮助实现团队合作。GitHub是代码合作平台，我们撰写的代码能够同步到GitHub上方便所有人了解代码进度。Leangoo主要用来实现我们看板的管理和燃尽图的自动绘制，通过它我们能够每日更新我们的任务进度和功能完成情况，使我们对项目计划能够作出及时调整和推进。

1. **什么功能产生的Bug最多，为什么？在发布之后发现了什么重要的bug? 为什么我们在设计/开发的时候没有想到这些情况?**

主要是在按不同条件搜索部分产生的Bug较多，在拉取消息和提交筛选条件时提交和返回状态会出现与结果不符的情况。

1. **代码复审（Code Review）是如何进行的，是否严格执行了代码规范？**

  代码复审我们主要是在测试阶段各自检查负责的代码，查找是否有报错或其他不符合规范之处。我们在开发前就规定了我们开发时候需要遵循的代码规范，因此在开发过程中我们基本都有按照规范来执行。但没有对代码进行一次完整的审查，这是我们的不足之处。

**测试/发布**

1. **团队是否有一个测试计划？为什么没有？**

前期根据我们程序的分析、调查、功能的需求分析，我们团队有定制测试计划，计划如下：

Alpha阶段测试计划

在冲刺的最后两天负责测试，前期完成功能时记录出现的bug

正因为制定了较详细的测试计划，也使得团队项目的有序进展，使得项目能够更加落地，符合预期所想，保障在用户的使用过程中，避免在用户使用的过程中出现bug。尽可能的把用户的所有使用情况都模拟到位。

1. **是否进行了正式的验收测试？**

我们仅在本机上针对写的测试用例一个一个演示测试实现的功能。我们从管理用户角度进行测试，确保用户可以正常使用拥有的功能。我们实现了功能测试、兼容性测试以及UI测试。在测试过程中，我们将记录每个测试点的测试结果和问题，并及时与开发团队沟通，以便及时修复和迭代。最终，通过持续不断的测试和优化，确保产品能够满足用户需求并提供良好的用户体验。并且我们预计在Beta阶段进行压力测试。应该算不是非常完备的“验收测试”。

在具体测试中，我们将实现以下场景和测试用例：

1. 查看个人信息

场景描述：用户点击个人信息。

测试用例：检查用户的基本信息是否正确显示；修改个人信息后，检查修改结果是否保存成功。

2. 登录

场景描述：用户输入账号密码，点击登录按钮。

测试用例：输入账号和密码，检查登录是否成功；尝试使用错误账号登录，检查是否无法登录。

3. 注册

场景描述：用户输入注册的账号和密码。

测试用例：检查注册功能是否正常；注册是否成功并能够登录。

4. 查看员工信息

场景描述：用户在界面上查看员工信息列表。

测试用例：确认用户查看的信息完整正确。

5. 修改员工信息

场景描述：用户点击修改按钮进入修改页面并对信息进行修改。

测试用例：检查原本的信息是否准确，并确认修改结果是否正确。

6. 发布应聘消息

场景描述：用户点击添加按钮进入消息空白页输入岗位信息并发布。

测试用例：检查发布信息是否正确。

1. **团队是否有测试工具来帮助测试？**

在alpha阶段由于尚未进行压力测试，因此我们团队使用了AndroidStudio进行测试。Android Studio 是谷歌为其 Android 操作系统开发的官方集成开发环境 (IDE)，基于 JetBrains 的 IntelliJ IDEA 软件，专门用于 Android 应用程序开发提供全面的调试功能，包括内存、CPU、GPU 和网络分析工具，帮助开发者优化应用性能。支持在设备上直接进行调试，查看实时数据。

1. **团队是如何测量并跟踪软件的效能的？从软件实际运行的结果来看，这些测试工作有用么？应该有哪些改进？**

只在本机上进行了测试，未进行测量并跟踪软件。从软件实际运行的结果来看运用测量和跟踪软件是有一定成效的。能够提早找出问题并能及时解决。

**5. 在发布的过程中发现了哪些意外问题？**

除了一些常规的代码逻辑bug之外，我们再发布时由于服务器端部署比较复杂，涉及到一系列数据库操作，恢复数据操作，改变结构操作，总体来说没有什么意外的风险。

二、团队分工与成员贡献

根据团队成员在Alpha阶段的角色和具体贡献排序：

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 角色 | 团队贡献排序 | **可验证的贡献** |
| 黄文辉 | PM | 1 | 登陆注册、信息展示、筛选搜索、信息修改等功能的控制类、实现类，看板及燃尽图管理 |
| 郑玮 | 测试 | 3 | 整个项目的运行测试、测试用例和测试报告的完成 |
| 代远军 | 前端开发 | 2 | 登陆注册、信息展示、筛选搜索、信息修改、等功能前端界面进一步美化 |
| 李勇杰 | 前端美工 | 4 | 登陆注册、职位展示、条件搜索、数据展示、个人信息界面等功能前端美工设计 |
| 付浩帆 | 后端开发 | 5 | 登陆注册、信息展示、筛选搜索、信息修改等功能的逻辑接口类 |
| 夏佳雨 | 数据挖掘 | 6 | 数据库设计及搭建、参与实体类设计，学习API文档，获取服务器端数据 |

贡献排行如下：

黄文辉：25

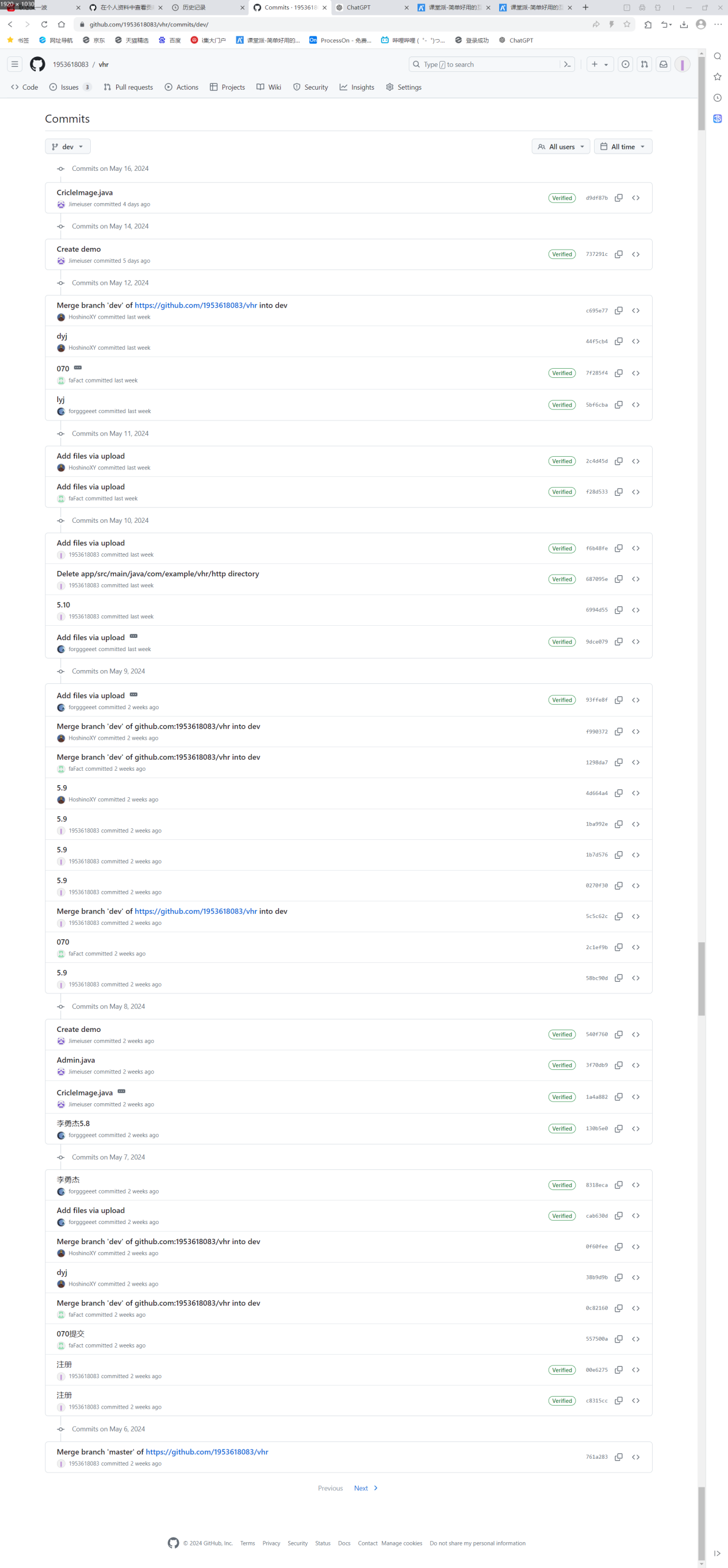
代远军：23

郑玮：21

李勇杰：19

付浩帆：17

夏佳雨：15



## 每个人请思考：你在该团队是否发挥了已有的水平或者需要更换团队？如果需要更换，请自行选择其他团队，其他团队有权决定是否接收。

我们团队所有人均发挥了已有的水平，团队气氛及其融洽，并不更换团队。

三、报告按时提交课堂派