**高校社交化编程及代码托管平台的方案构建及其实现**

**摘 要：**针对高校内软件开发大环境中还没有可行的校内代码托管平台的局限，本文提出了一套完善的、可行的、开放的基于在线代码托管服务打造的技术协作与分享平台方案，该方案使用分布式版本控制系统Git，基于开源项目GitLab开发，提供项目管理、代码托管、在线编辑、技术交流分享等社交化编程功能。

关键词：编程；协作；社交

**Abstract:** This paper put forward a proposal of technical collaboration and sharing which is the solution of there is no feasible code hosting platform among many colleges and universities. This scheme is perfect, feasible and open. It is base on the service of source code hosting. It uses Git -- a distributed version control system and it is based on the development of an open source project GitLab. At the same time it provided project manage, code hosting, online editing and technology sharing, these functions are conducive to social programming.

**1 引言**

今时今日，每个行业都在苦苦地追寻缩短盈利周期的道路，软件产业也不例外。由于软件行业的技术更新迭代迅速，引入新技术、降低成本预算、缩短盈利周期等因素的共同影响下，迫使软件行业的精英们寻找一种能够破除这种困境的解决方案，因此催生了代码协作、版本控制等概念。

广州大学华软软件学院，作为一所本科层次、主打培养软件人才的独立学院，在软件工程方面暂时还没有一套完善的、可行的、开放的代码托管方案，另外在软件开发协作方面力量比较分散，有强烈的代码托管和版本控制网站需求。本文对此提出了一套便捷高效的校内公共分布式版本控制解决方案与创新优势。

**2 版本控制**

在一套软件开发流程中，文档/代码的编写/修改是尤为重要的。尤其是在当今流行的敏捷开发中，快速迭代、代码重构、渐进增强等步骤都是必经的几个阶段。而在这个过程中，软件工程师们为了追踪、维护源码、文件等改动，都必不可少地使用了版本控制系统。通过版本控制，你可以找出问题出现的原因，对比文件的细微变化，将文件回溯到之前的某个状态，甚至可以将所有东西都回滚到某个时间点，这一切都能交给版本控制系统去轻松解决。目前最流行、最先进的版本控制系统，非Git莫属。

Git是一个分布式版本控制/软件配置管理软件，与CVS、Subversion一类的集中式版本控制工具不同，它采用了分布式版本库的作法，不需要服务器端软件，就可以运作版本控制，使得源代码的发布和交流极其方便。Git的速度很快，最为出色的是它的合并追踪能力。

**3 社交化**

在线协作。遵循“Fork + Pull”模式，项目的参与者们只需要直接在项目页面上直接点击“Fork按钮”去派生这个版本库，而不需要向项目所有者申请提交的权限，大大地增加了项目传播与协作预备的效率。参与者在对项目修改并提交后，可以先自行推送到自己的派生版本库，然后再通过向原项目发送Pull Request的方式让原项目所有者/维护者进行拉取/合并，合理地避免了因为等待原项目合并而产生的无法继续提交等各种问题。相比其他的协作模式，这种分享提交的在线协作模式将明显有助于保证工作流的流畅性、可靠性和一致性，适用于小型团队开发和第三方开发者向开源项目作出贡献。同时，本文所提出的系统也具有在线编辑功能，方便开发者随时为项目作出简单的修改。

项目讨论。项目协作中，讨论交流是必不可少的一部分。此平台提供了一个高效的交流沟通途径，帮助用户为项目随时创建交互式、可持续的对话，以便对项目中的bug、新功能、疑问或者其他想法的交流，简单地降低了开发者之间的沟通成本，节省了因为电子邮件沟通而产生的不必要的开销，也把项目的有效讨论跟项目自身紧紧地融合到了一起，为后来的开发者们提供更多的参考。小巧易用的沟通体系也与上文提到的Pull Request紧密整合，是协作工作流的有利补充。

团队管理。上文提到的Pull Request适用与第三方开发者为开源的公开项目作出贡献，而在固定的开发团队中则有更好的协作方法。用户通过创建“组织”的方式在平台上组建团队，邀请团队成员之后就能开始新项目的开发工作。这一功能满足了开发团队中合作困难、管理混乱的问题，也能在同一组织中划分不同的项目团队，适合不同规模的团队/组织进行团队协作。同时也在这基础上构建了不同的权限体系，管理人员只需轻松的几下点击就能完成授权操作，保证了项目和版本库的安全性。

社交互动。年轻人喜欢在网络中互动，程序员们也不例外。作为一个高校社交编程与代码托管平台，这个功能当然是亮点所在。平台中提供对项目和对个人的两种互动模式，对于项目提供添加收藏、提交评论、关注等功能，对于个人开发者也将关注、互加好友、即时聊天、查看公开活动时间线等流行的互动功能通通收入囊中。

**4 示例**

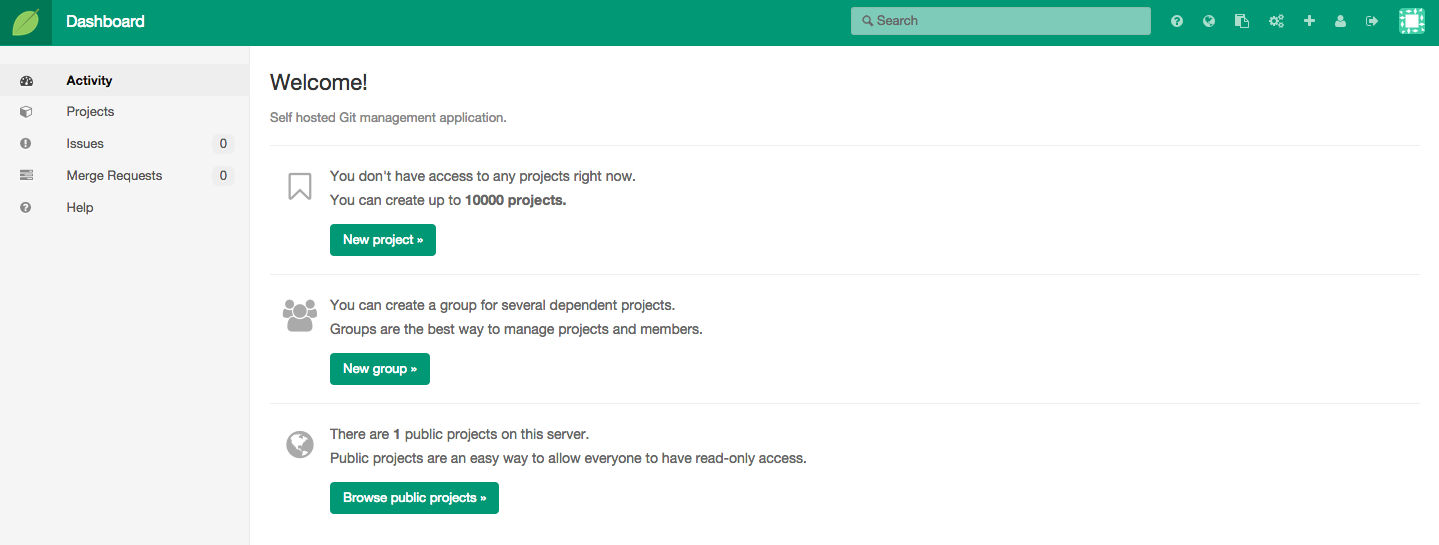


图1 首页

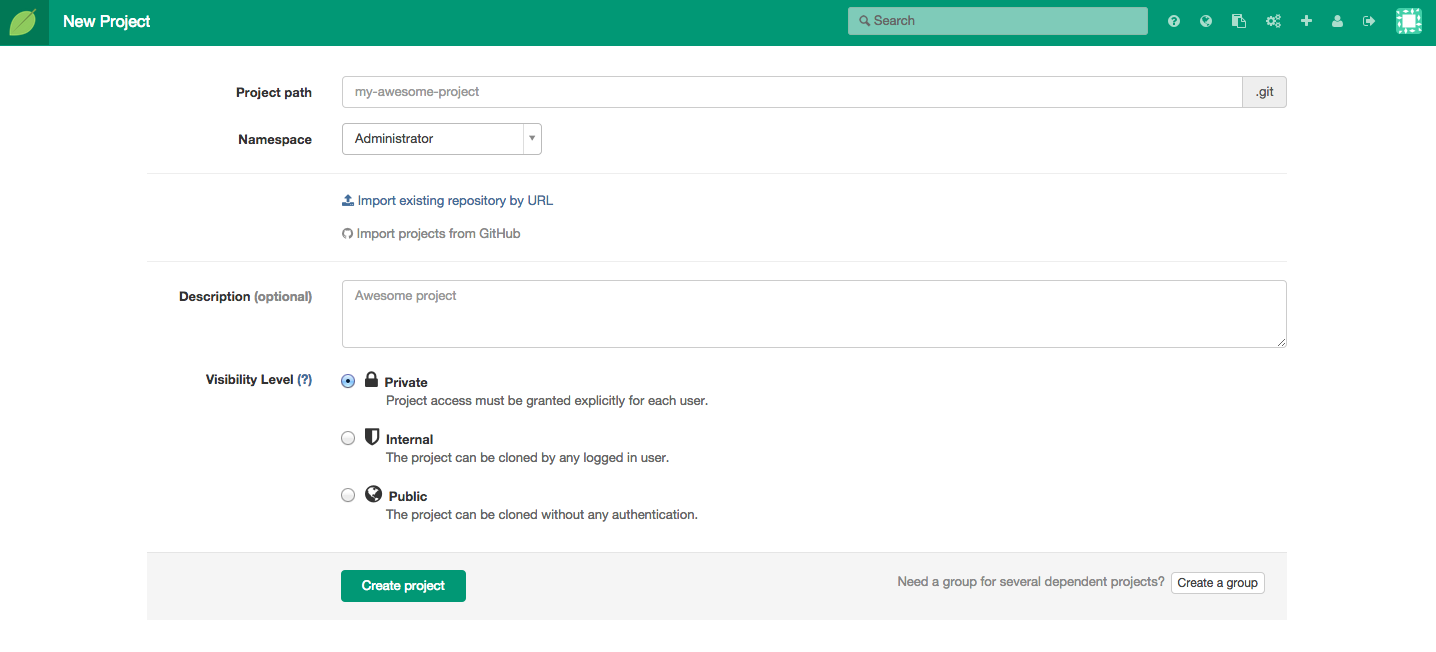


图2 新建项目

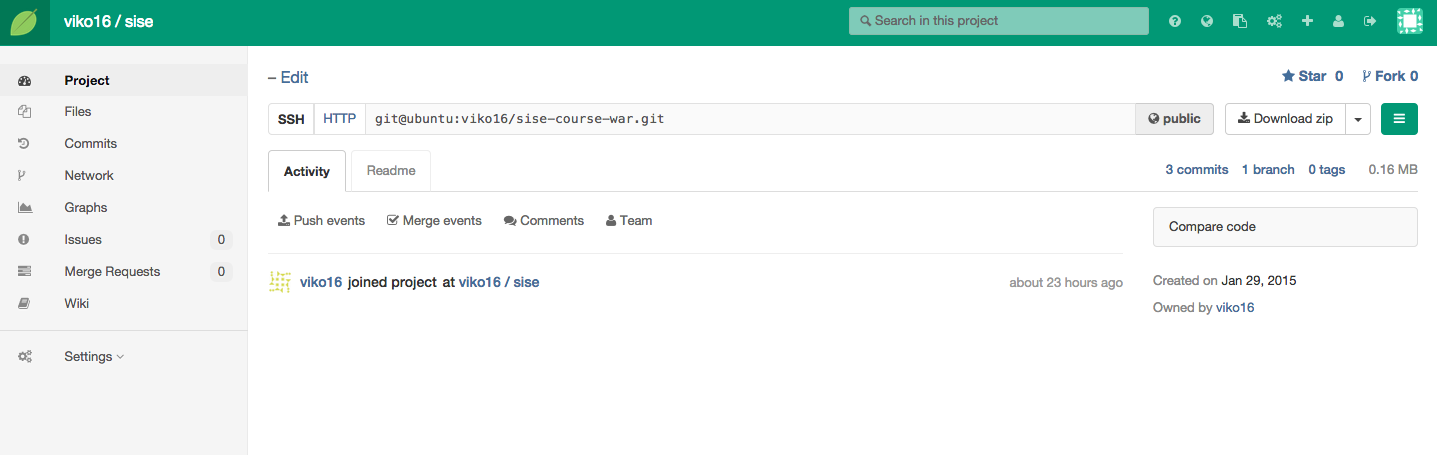


图3 项目页面

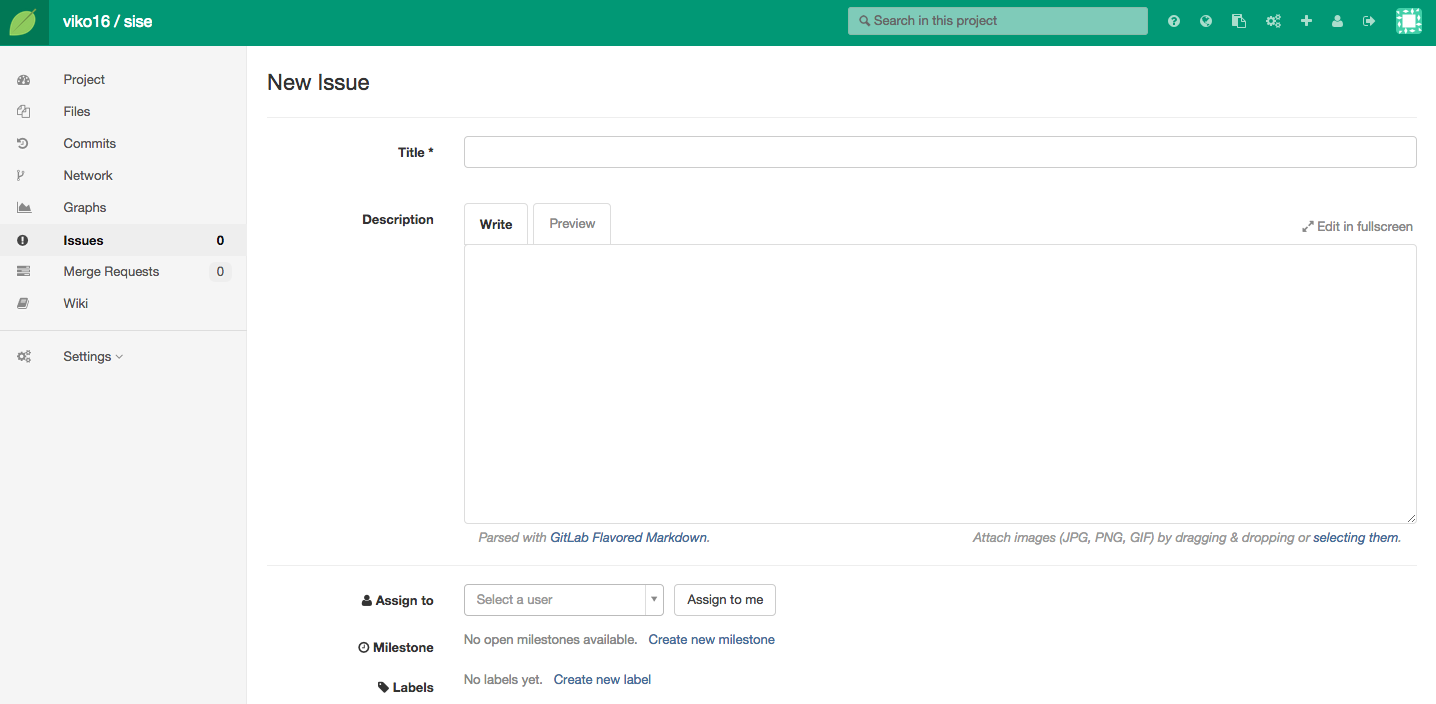


图4 新建讨论

**5 结论**

本文提出的平台方案，适用于各高校内部署，这样既能发挥校内局域网的优势将校内师生的优秀项目聚集在一起，也能有效地提高师生们的代码协作能力和促进技术经验分享氛围。对于用户，能够降低项目协作成本，让贡献代码变得更轻松更容易。对于优秀的开源项目，将是一个很好的展示、分享平台，有助于项目的推广与壮大。去中心化的模式，弥合了项目创作者与目标人群之间的裂缝，降低了人与人之间沟通成本，让项目更富有生命力。

下一步要做的工作：继续完善系统稳定性，以及在用户可见的浏览器端上增加更多人性化的功能。

参考文献

**[1]** Pro Git

[2] GotGitHub