“交你办”项目需求分析文档

1. **需求获取**

本项目的范围是开发一个学生交流互助平台，主要包括学生发帖提问、回答和学生发布委托、承接委托的功能，并将委托同步到时间轴上显示。该平台旨在促进学生之间的问题解答和组队合作，提供一个方便的交流平台。

1. **可行性分析**

技术可行性：

开发平台：基于Django开发的项目。Django是一个功能强大的Python Web框架，提供了丰富的功能和工具，适用于构建学生交流平台这样的应用。

编程语言：使用Python作为主要编程语言。

数据库： Django提供了对多种数据库的支持，如MySQL、PostgreSQL和SQLite等，可以根据项目的需求选择MySQL数据库系统。

界面设计：使用Django的模板语言和前端技术（如HTML、CSS和JavaScript）来设计用户界面，实现学生发帖提问、回答和发布、承接委托等功能的展示和交互。

操作可行性：

小项目规模：考虑到项目规模为五人小团队，开发和操作可行性较高。小团队可以更加灵活和高效地协同开发，减少沟通成本和管理复杂度。

学生发帖提问和回答功能：通过设计用户友好的界面和交互流程，学生可以轻松地发表帖子、提问问题和回答他人的问题。可以提供适当的编辑器和格式化选项，以便学生能够按需编辑和格式化他们的帖子内容。

学生发布委托功能：提供一个表单或界面，学生可以填写委托的相关信息，如项目名称、描述、成员要求等。发布后，委托邀请会同步到时间轴上显示，方便其他学生浏览和加入。

1. **相关限制和约束**

技术限制：项目基于Django开发，开发团队熟悉Django框架和相关技术，并具备良好的Python编程能力。

时间限制：项目在规定的时间范围（两周）内完成，团队合理规划开发和测试的时间，并确保按时交付。

资源限制：项目的开发和运行适当的硬件设备、网络资源和数据库等支持。团队评估和获取所需的资源。

用户体验：项目提供良好的用户界面和交互体验，以满足学生的需求和期望。

数据保密：在处理学生的帖子和信息时，团队确保数据的安全性和保密性，遵守相关法律法规。

1. **需求定义**
   1. 引言

本文档旨在描述软件项目“交你办”（以下简称项目）的需求分析。该项目旨在开发一个学生交流互助平台，提供学生发帖提问、回答和学生发布委托，承接委托的功能，并将委托邀请同步到时间轴上供其他学生浏览。以下将详细描述项目的需求和相关信息。项目的主要用户为学生群体，包括大学生、研究生等。他们可以使用该平台进行问题提问、回答和参与委托承接。

在拓展性上，本项目与学校教育系统在未来可能存在一定的关联，以便获取学生信息和课程信息等数据。

* 1. 功能需求

学生发帖提问：学生可以在系统中发布问题，包括问题的标题和详细描述。系统能够及时展示发布的问题。

回答问题：学生可以浏览问题列表并选择合适的问题进行回答。系统能够记录回答的内容，并及时显示在相关问题下。

学生发布委托：学生可以发布委托，包括组队项目的名称、描述、要求和联系方式等信息。系统能够记录并展示委托的内容。委托同步到时间轴：系统能够将发布的委托同步到时间轴上，以便其他学生浏览和参与。

个性化推荐：系统根据学生浏览帖子的情况，将学生主页的帖子进行个性化排序。

* 1. 性能需求
     1. 存储容量限制：系统能够处理和存储大量的问题、回答和发布委托数据，具备足够的存储容量。
     2. 执行速度和响应时间：系统具备快速的执行速度和响应时间，以保证用户操作的流畅性和高效性。
     3. 吞吐量：系统能够支持同时处理多个用户的操作请求，并具备较高的吞吐量。
  2. 环境需求
     1. 硬件设备：系统在支持现代网页浏览器的计算机或移动设备上运行。
     2. 软件操作系统：系统能够在主流操作系统（如Windows、macOS、Linux）上运行。
     3. 网络和数据库：系统与互联网连接，并具备与数据库进行数据存储和检索的能力。
  3. 用户因素
     1. 输入和输出：系统接收学生发布问题、回答和委托的输入，并展示给其他用户进行浏览和参与。同时，系统根据学生对问题帖子的浏览情况进行不同主题的排序，在主页进行个性化推荐。
     2. 数据格式：系统对输入和输出的数据格式有一定的规定，以确保数据的一致性和正确性。
     3. 数据存储介质：系统选择适当的数据库来存储用户数据和时间轴信息。
  4. 文档需求
     1. 文档：用户手册、管理员手册、开发文档、数据库设计文档等。
     2. 文档针对的读者：用户手册针对学生用户，管理员手册针对系统管理员，开发文档针对开发人员。
  5. 数据需求
     1. 输入输出数据格式：系统规定问题、回答和委托的数据格式，以保证数据的一致性。
     2. 数据频率：系统支持频繁的数据输入和输出操作，能够快速响应用户的请求。
     3. 数据准确度和精度：系统确保输入和输出数据的准确性和精度，避免错误和误解。
     4. 数据流量和保持时间：系统处理大量的数据流量，并能够保持数据的合理时间范围。
  6. 资源需求
     1. 运行资源：系统足够的计算资源、存储资源和网络资源来保证系统的正常运行。
     2. 软件开发或维护所需的人力：系统的开发和维护具备相关的人力资源，包括开发人员、测试人员和运维人员等。
     3. 支撑软件和开发设备：系统开发过程中可能使用支撑软件（如开发工具、版本控制系统等）和适当的开发设备（计算机、服务器等）。
  7. 安全保密需求
     1. 访问控制：系统实施适当的访问控制措施，确保只有授权用户可以访问和操作系统。
     2. 数据隔离：系统能够隔离不同用户之间的数据，确保数据的私密性和安全性。
     3. 系统备份要求：系统定期进行数据备份，以防止数据丢失或系统故障。
  8. 软件成本消耗与开发进度需求
     1. 第一周：完成基本功能，包括发表问题、回答问题，并发布委托及时间轴
     2. 第二周：完成日志系统、登陆界面和申请界面，完善个性化定制功能
     3. 软件成本消耗：无
  9. 质量保证
     1. 可靠性要求：系统具备高可靠性，能够稳定运行并提供准确的功能。
     2. 错误监测与隔离：系统能够及时监测和识别错误，并采取相应的隔离措施以避免影响系统的正常运行。
     3. 平均出错时间和重启时间：系统具备较长的平均出错时间，并能够快速恢复和重启。
     4. 改进和可移植性：系统具备可持续改进的能力，并具备一定的可移植性，以便在不同环境中部署和使用。

1. **数据建模**

数据对象：

学生（Student）：代表平台上的学生用户。

属性：用户ID、邮箱、密码、用户名等。

帖子（Post）：学生发布的帖子或问题。

属性：帖子ID、标题、内容、发布时间等。

回答（Answer）：学生对帖子的回答或解答。

属性：回答ID、内容、回答时间等。

关系和联系：

学生与帖子之间存在一对多关系（一个学生可以发布多个帖子，一个帖子只属于一个学生）。

帖子与回答之间存在一对多关系（一个帖子可以有多个回答，一个回答只属于一个帖子）。

学生与回答之间存在一对多关系（一个学生可以发布多个回答，一个回答只属于一个学生）。

具体建模如下图所示。

