**南京理工大学计算机科学与工程学院**

**软件工程专业毕业实习报告**

**班 级**

**学生姓名**

**学 号**

**实习时间**

**指导教师**

**南京理工大学计算机科学与工程学院制**

|  |
| --- |
| 一 毕业**实习简况**  **1.1实习时间和地点**  实习时间：8.26-9.12  实习地点：华为南京研究所、南京理工大学  **1.2实习目的**  具体实习目的：  熟练掌握快应用的开发及开发规范，小组合作完成一个具有实际意义的快应用。  **1.3任务及要求**  开发一个快应用，实现用户注册登录功能，使用户能够修改自身用户信息，每个用户可以关注其他用户，提供用户图片上传、下载、评论、点赞、转发、修改上传图片等功能，能够调用华为的AI进行图片的识别分类，供用户按照标签进行搜索，设计一个推荐算法，根据通过统计用户关注的用户的共同关注用户并建立一个有序列表，向用户按序推荐。此外，每当用户关注一位用户，重新进行统计，并在下方进行用户推荐。  **二 项目实习技术总结**  1概述  开发一个快应用，实现用户注册、登录功能，登录后，主要界面分为三部分，分为个人信息页面，社区页面，以及排行榜页面。个人信息页面可以让用户查看和修改个人信息，查看当前粉丝，按用户名搜索其他用户；排行界面主要分为点赞排行以及美学评分排行，其中点赞排行会根据社区中用户的点赞行为实时更新，美学评分排行榜，是调用AI engine对上传的图片进行美学评分，在美学排行榜中进行更新；社区界面可以让用户浏览图片，按标签搜索图片，对于图片可以进行评论，或者对于他人的评论进行回复，实现推荐算法，用户可以关注感兴趣的其他用户。  2需求分析  2.1 功能需求分析  实现注册功能，使用邮箱或手机号验证，注册完成将用户信息存入个人信息数据库，用户登录时，向后端数据库验证用户名密码，验证成功即可登录，登录后可以在个人信息界面中查看自身的信息，查看自己上传的图片，修改自身信息，查看自己的具体有哪些粉丝。    登录注册  实现社区功能，浏览上传的图片，对图片进行评论，或者回复他人评论，也可以对于图片进行点赞。  实现两个推荐算法，动态推荐，即在浏览图片时，会弹出一个你可能感兴趣的提示框，显示排行榜以外的高评分或者高点赞图片，同时还有用户推荐算法，用户推荐算法通过统计用户关注的用户的共同关注用户建立一个有序链表，向用户推荐其他用户，每当用户关注一位其他用户时，都会实时更新此链表。在用户选择关注其他用户之前，该用户可以看到其他用户目前正在被多少用户关注。  实现图片排行，包括点赞数量排行以及美学评分排行，美学评分需要调用AI进行评分，在将得到的分数进行排行，在用户每次上传图片的时候，对上传图片进行美学评分，并实时更新美学评分排行榜；点赞数量由社区用户进行点赞，统计数量，更新排行榜。  实现发现的界面设计，主要包括搜索栏以及照片墙，照片墙上推荐的是数据库中权重大于一定阈值的图片（主要考虑排行榜有时限，这里的推荐可以推荐以前的图片）。照片墙只展示单张图片，当用户对照片墙中的某张图片感兴趣时可以点击该图片，随后进入单个动态子模块提供支持的动态详情界面，该界面显示单个动态的具体内容，用户也可以在此进行点赞转发和评论操作。然后上方搜索栏点进去有账号推荐和标签推荐，然后有实时搜索功能。  实现发布动态的功能，设计一个发布动态的单独tab，点进去可以从相册里选择图片，也可以把已经选择的图片删除，也能做点简单的编辑。  未命名文件    2.2 对性能的需求分析  对应用性能需求主要的是使用频率较高，在应用中关键程度非常高的部分，以及资源占用非常严重的部分。性能需求主要包括，（1）响应速度：API请求的平均时间，登录响应时间等（2）服务正常运行时能够支持的最大在线人数（3）应用中的某一个接口可以同时被多少用户调用（4）在用户在线人数超过预期的人数时，应用提供的服务有多少可以继续提供（5）当实际的的运行压力超出预期的与运行压力时，应用可以稳定运行多久（6）系统能支持多少虚拟用户（即每天登陆应用的人数）。  对于性能的具体需求主要依据以下几个方面获取：（1）由用户提出明确的对于性能的需求分析（2）依据已有的数据（日志文件）进行分析，得出具体性能需求（3）根据同类型服务应用的数据得出具体性能需求。  我们小组的快应用的需求主要是注册时，使用手机或者邮箱进行验证时，验证信息需要快速的发送给用户，登录时需要迅速地向数据库确认用户名密码，并给出登录成功或者用户名密码错误的信息；在使用服务时，对于接口的调用的请求的响应时间控制在一个很短的时间内；在用户上传图片之后，马上调用AI engine 对于图片进行美学评分，更新排行榜；在多用户同时在线时，服务能够稳定地提供给每一位用户。  2.3 输入输出要求  输入设计的目标：保证向系统输入正确的数据。在此前提下，尽量做到输入方法的简单、快捷、方便。  输入内容就是用户上传的图片，输入方式是在社区中进行直接上传图片，输入的要求是在上传时，可以在手机相册中选取，也可以直接拍照上传，但是必须是图片的格式，而且再上传时，设备必须连接网络，否则会上传失败。  在进行输入之后，后端进行中间处理，调用AI engine 对图片进行美学评分以及标签分类，为图片添加一个URL，将图片以及图片的信息使用腾讯云对象存储COS，依赖于腾讯云对象存储cos技术的断点续传、自定义对象头部、静态网站、数据迁移等功能高效便捷的完成图像存储系统的设计与搭建；对图片添加点赞接口、获取点赞数量接口（整合至时间线接口，该接口不能被调用）、取消点赞接口、评论接口、获取评论数目接口（整合至时间线接口，该接口不能被调用）、获取评论内容接口、取消评论接口。中间处理结束之后就是输出。  输出设计的重要性：评价管理信息系统是否能为用户提供准确、及时、适用的内容的标准之一。输出设计的内容：确定用户在使用信息方面的要求包括使用目的、输出速度、使用频率、安全性等要求。  输出的内容就是将经过中间过程处理的图片上传到社区，每一种功能图片会占据手机的全屏幕，以便用户可以清晰的看到全图，下部显示部分评论，点击可以查看详细的评论；此时图片已经添加了部分接口，点赞与评论的数据也会通过接口传入图片数据库中。    2.4 模块要求分析  本应用主要分为以下几个模块：（1）个人模块   1. 时间线模块 2. 动态管理模块 3. 点赞管理模块 4. 评论管理模块 5. 关注管理模块 6. 推荐模块 7. 搜索模块   2.4.1 封闭性  开放封闭原则是所有[面向对象](https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1/2262089" \t "https://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E5%B0%81%E9%97%AD%E5%8E%9F%E5%88%99/_blank)原则的核心。软件设计本身所追求的目标就是封装变化、降低耦合，而开放封闭原则正是对这一目标的最直接体现。关于开放封闭原则，其核心的思想是：软件实体应该是可扩展，而不可修改的。也就是说，对扩展是开放的，而对修改是封闭的。因此，开放封闭原则主要体现在两个方面：对扩展开放，意味着有新的需求或变化时，可以对现有代码进行扩展，以适应新的情况。对修改封闭，意味着类一旦设计完成，就可以独立完成其工作，而不要对类进行任何修改。  对修改封闭的有两个例外：一是对于缺陷修复的改动，缺陷在软件中很常见，是不可能完全消除的。当缺陷出现时，就需要我们修复现有的代码。软件修复明显倾向于实用主义而不是坚持开放封闭原则；二是对客户端无影响的修改，对于客户端无影响部分即松散耦合，这些类的变化是不会引起别的类的改变的，所以修改这些类对于用户来说没有直观的变化。    接口  在我们小组的项目中封闭性主要体现在了，对于用户所有能使用的功能，都提供了一个调用该功能的接口，用户的动作完全由接口来实现，每设计一个新的功能，就为该功能设计一个接口，供用户调用，而用户程序不需要任何的修改。对于上述的8个模块，每个模块的每一个基础功能都需要用户调用接口实现，模块的变化对于客户端没用影响。每当需要开发一个新的功能时，只需要完成功能实现代码以及一个调用的接口，在客户端直接调用即可。    2.4.2 安全性  安全性设计主要包括两方面，一是数据库的安全设计；二是接口调用过程中的安全设计。  数据库的安全性设计：数据库的不安全因素主要是（1）无授权用户对于数据库的恶意存取以及破坏（2）数据库中重要的信息的泄露（3）安全环境的脆弱。我们采取的措施有用户身份鉴别，存取控制，授权，加密存储。关于上述的第三个安全问题主要是系统的安全问题，不作过多讨论。  为了应对无授权用户对于数据库的恶意存取以及破坏问题，主要采用的方法是用户身份鉴别以及存取控制，用户身份鉴别可以是静态口令鉴别，类似于设置一个密码，或者动态令牌的形式进行鉴别；存取控制机制主要包括定义用户权限，并将用户权限登记到数据字典中和合法权限检查。两者机制一起组成数据库管理系统的存取控制子系统。针对于数据库中重要的信息的泄露问题使用的安全方法主要是数据加密存储。  接口的安全性设计：在访问接口的过程中如果没有过滤，可能会出现安全性问题，例如通过模拟请求，对数据进行篡改，因此需要设计一个安全的接口调用方式。具体的方案可以是（1）使用token串验证，每当用户登录时，服务器生成一个唯一的token串给用户，在请求调用时，可以通过验证token串来确保安全性。（2）使用时间戳验证，每次请求发出是可以在请求上附上一个时间戳，在接收端设置一个检测如果超时，就拒绝请求（3）使用token串与时间戳相结合并进行加密的签名机制。接收端写一个过滤器，只有token串有效，时间戳未超时，缓存服务器中不存在签名同时满足，才应答此次请求。  安全  2.4.3 容错性  容错性设计原则：（1）提示与引导，包括尽可能详尽的引导与文字提示，且文字提示的内容简单易于理解，进行不可逆的操作时向用户询问是否继续；（2）限制操作，主要包括加大不可逆操作的难度，限制用户的某些不必要操作，减少出错可能（3）反馈与帮助，是指在错误发生后能够让用户及时反馈错误并提供纠错帮助，错误提示信息需要易懂且精炼（4）错误恢复，能够帮助用户在错误发生后快速回到安全状态。（5）减少负担，主要是减少用户不必要的操作步骤。  容错性  结合上述的容错性设计原则，容错性设计可以分为以下三个阶段：引导、报错、解决。  引导主要针对新手用户，让他们很快的了解设计出的应用的核心功能以及主要的操作方式，帮助他们更快的上手，当然引导功能的实现方式不仅限于此，输入框中的输入提示也是常见的引导样式。输入框是用户完成信息录入的主要途径之一，有录入才有报错，有报错才需要引导。引导归根结底是为了避免用户在操作过程中犯错，而用户的操作过程又可以看成是不断做决策的过程，要想做出正确的决策必须要消减信息的不对称性。所以引导功能我们可以理解为就是消减信息的不对称性，让用户做出正确的决策。  报错就意味着引导失效，用户还是错误的使用了某一功能，报错主要分为报错方式以及报错时机，报错方式主要就是弹框，而报错的时机就比较重要了，有些错误需要请求数据库或者服务器才可以发现，但是一些简单的可以直接由前端界面进行判断的错误，如果依然滞后报错可能不太合适。  解决即向用户提供纠错的帮助，需要注意的是在给用户提供解决方案的时候，尽量提供跳转链接，方便用户操作。  我们的快应用中错误可能发生的场景包括：注册时的用户名重复，登录时的用户名或者密码错误，修改个人信息时，用户输入的手机号错误，各种需要使用网络的时候，网络连接错误，搜索过程中未找到搜索内容，还有就是可能存在的服务器超载问题等。  对于上述可能发生的问题都会进行反馈，用户根据应用给予的反馈，通过反馈的可能问题尝试解决。对于服务器的超载问题在不升级服务器的情况下，解决方案可以是（1）在服务器中直接形成请求的处理进程，不撤销处理进程，所有的请求成队列等待处理，这一解决办法的问题是响应速度慢，而且在有时间戳的情况下可能会请求失败（2）使用相对于进程而言开销较小的线程来处理请求。 |