

Valet需求分析规约

1652772 肖 睿

1652712 赵欣宇

Valet需求分析规约

○ 修订历史

一 引言

- 1.1背景
- 1.2参考资料
- 1.3假定和约束
- 1.4用户的特点

二 功能需求

- 2.1系统范围
- 2.2系统体系结构
- 2.3系统总体流程
- 2.4需求分析
 - 2.4.1 功能模型—活动图
 - 2.4.2数据模型--分析类图
 - 2.4.3行为模型--序列图

三 非功能需求

- 3.1性能要求
 - 3.1.1精度
 - 3.1.2时间特性要求
 - 3.1.3输入输出要求
- 3.2数据管理能力要求
- 3.3安全及保密性要求
- 3.4灵活性要求
- 3.5其他专门要求
 - 3.5.1 可维护性
 - 3.5.2 可扩展性
 - 3.5.3 灾难恢复

四 运行环境规定

- 4.1设备
- 4.2支持软件
- 4.3接口

五 需求跟踪

○ 修订历史

编写日期	SEPG	版本	说明	作者	评审时间	评审参与人员	评审后修改批准日期	确认签字人员
2018.10.8	VA	0.1	初稿	赵欣宇	2018.10.8	赵欣宇 肖睿	2018.10.9	
2018.11.23	VA	1	定稿	赵欣宇 肖睿	2018.11.23	赵欣宇 肖睿	2018.11.24	
2018.12.3	VA	1.1	修订稿	赵欣宇 肖睿	2018.12.3	赵欣宇 肖睿	2018.12.4	

一 引言

1.1背景

随着社会的发展，人们生活水平大幅度提高，一方面人们所拥有的衣物数量大幅度增加，另一方面人们更加注重衣物的选择搭配，这就导致人们亟需一个高效便捷的衣物管理收纳方案。有很多人对衣物的关注度不高，拥有的衣物欠缺梳理，总是重复使用几件衣物，而衣柜中的大多数衣物在日常生活中的使用频率很低，在进行衣物整理时也缺乏有效信息，不知道该如何取舍衣物；还有人对衣物关注度较高，较为注重衣物之间的搭配以及各衣物的使用频率，需要一个记录穿搭和穿搭效果的平台。因此我们决定开发一个APP，提供衣物的线上数据存储和穿搭存储，并根据天气为穿衣提供方案，从而满足人们对衣物管理的需求。

1.2参考资料

[1] RESTful API – Wiki : https://en.wikipedia.org/wiki/Representational_state_transfer

[2] Metirial Design : <https://material.io/design/guidelines-overview/>

[3]Pressman R S. Software Engineering A Practitioner's Approach[M]. Seventh Edition. McGraw- Hill Higher Education, 2008.

1.3假定和约束

- 发布时间：2018年12月25日发布“Valet”智能衣橱APP测试版
- 系统要求：Android 7.1.1及以上版本
- 开发条件：成员自备开发使用电脑，并配置相关开发环境
- 开发效率：采用SCRUM敏捷开发模型，开发冲刺期间避免外界打扰。团队高度自治，队员们熟悉开发过程中涉及到的各种技术，紧密合作，确保每个迭代都朝着最高目标推进。
- 小组合作：为了加强小组之间的交流并且节约时间，除了线下讨论，还采用线上开会交流，讨论。
- 代码托管：为了充分实践软件工程的思想，对项目进行平台化、流程化管理。使用 github进行代码提交、合并、托管，产生最新最完整版的代码。

1.4用户的特点

本APP面向的人群为所有有衣物管理需求的，对衣物、穿搭管理存在一定困难或者想要借助数据的提高衣物管理水平、更有效的管理衣物。

用户应当每天在APP上获取穿搭建议，选取自己今日的穿搭，更新穿搭等。同时，用户需要在现实整理衣物时定期管理自己APP上的数据。

用户要为需管理的衣物拍照，并且通过APP上传到服务器保存起来。

二 功能需求

2.1系统范围

随着社会的发展，人们生活水平大幅度提高，一方面人们所拥有的衣物数量大幅度增加，另一方面人们更加注重衣物的选择搭配，这就导致人们亟需一个高效便捷的衣物管理收纳方案。有很多人对衣物的关注度不高，拥有的衣物欠缺梳理，总是重复使用几件衣物，而衣柜中的大多数衣物在日常生活中的使用频率很低，在进行衣物整理时也缺乏有效信息，不知道该如何取舍衣物；还有人对衣物关注度较高，较为注重衣物之间的搭配以及各衣物的使用频率，需要一个记录穿搭和穿搭效果的平台。因此我们决定开发一个APP，提供衣物的线上数据存储和穿搭存储，并根据天气为穿衣提供方案，根据用户喜好提供购衣推荐，从而满足人们对衣物管理的需求。

本系统应当以为用户提供数据化衣物管理、方便快捷的解决日常生活中会遇到的穿搭问题为出发点。

系统为用户提供衣物数据的线上存储，使用户数据更方便找回。

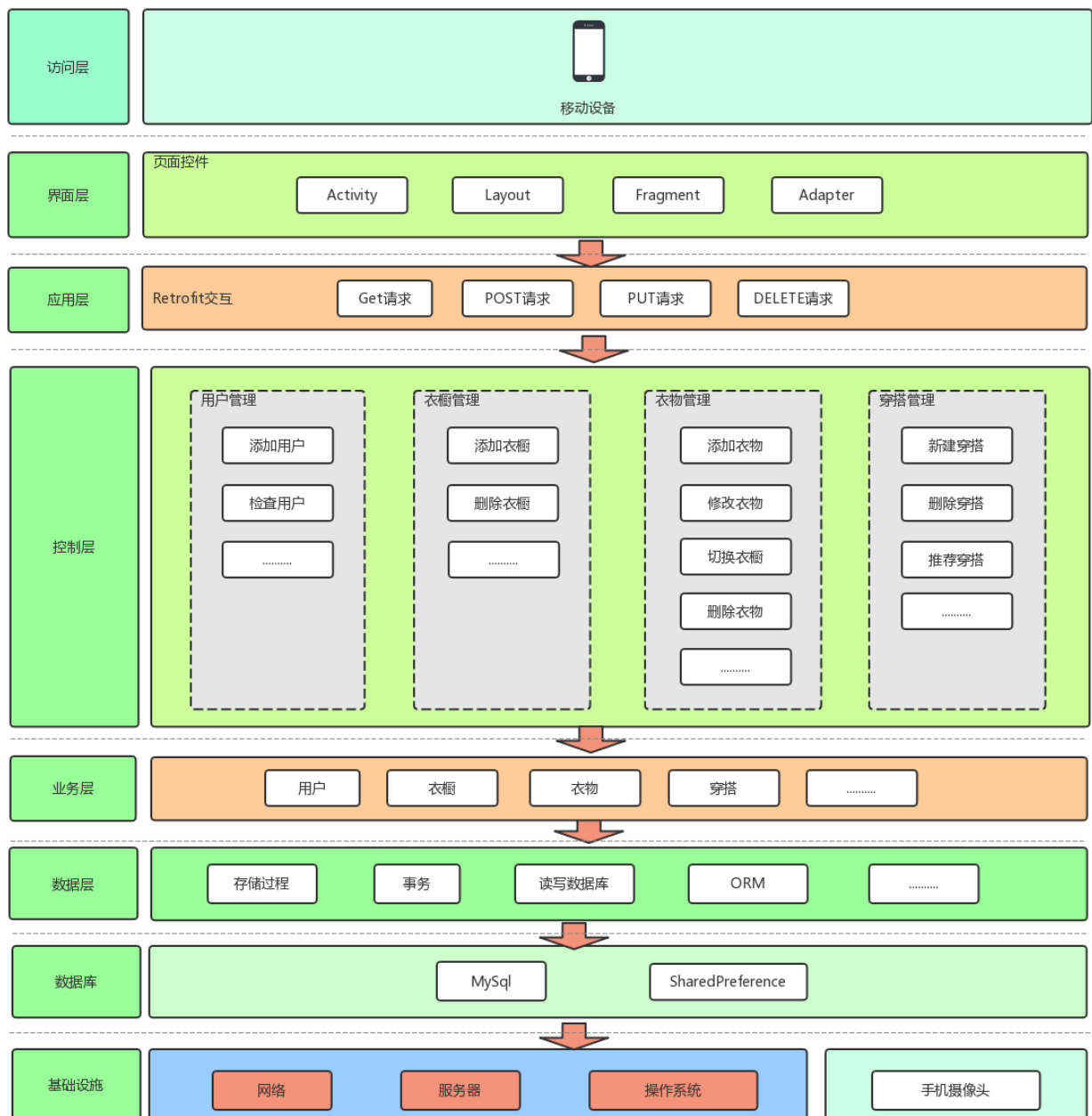
系统为用户提供人性化的推荐服务。根据当天的天气为用户提供穿衣推荐，根据用户现有的衣物为用户提供购衣推荐，用户还可以通过摇一摇解决选择穿搭时的选择困难问题。

系统规避了社交、好友互动等功能，使用户信息更加封闭化，增强衣橱的私密性。

2.2系统体系结构

本系统包括Android手机移动端和服务器端两个部分。手机移动端负责与用户之间的交互，呈现系统提供的内容信息，接收用户的操作请求并提供相应的反馈，这些操作请求包括从登录注册到衣橱、衣物管理的所有系统功能。服务器端为安卓端的信息展示提供内容和数据，接收安卓端操作请求并执行相应业务功能然后提供对应的内容反馈和数据响应。

系统的总体架构图如下。



2.3系统总体流程

用户进入APP时会先进入注册登录模块，首次进入APP时需要用户进行注册，登录时用户需要输入用户名和密码，如果选择保存密码系统会记住用户名和密码，方便下一次登录。

登录成功后进入系统主页，用户会在主页获得推荐的几个穿搭，用户可以摇一摇获得推荐穿搭中的随机一个穿搭，来帮助选择困难。主页面展示用户的所有穿搭，用户可以在主页面中的全部穿搭选一个穿搭作为今日穿搭。如果用户还没有穿搭会提示用户新建穿搭。

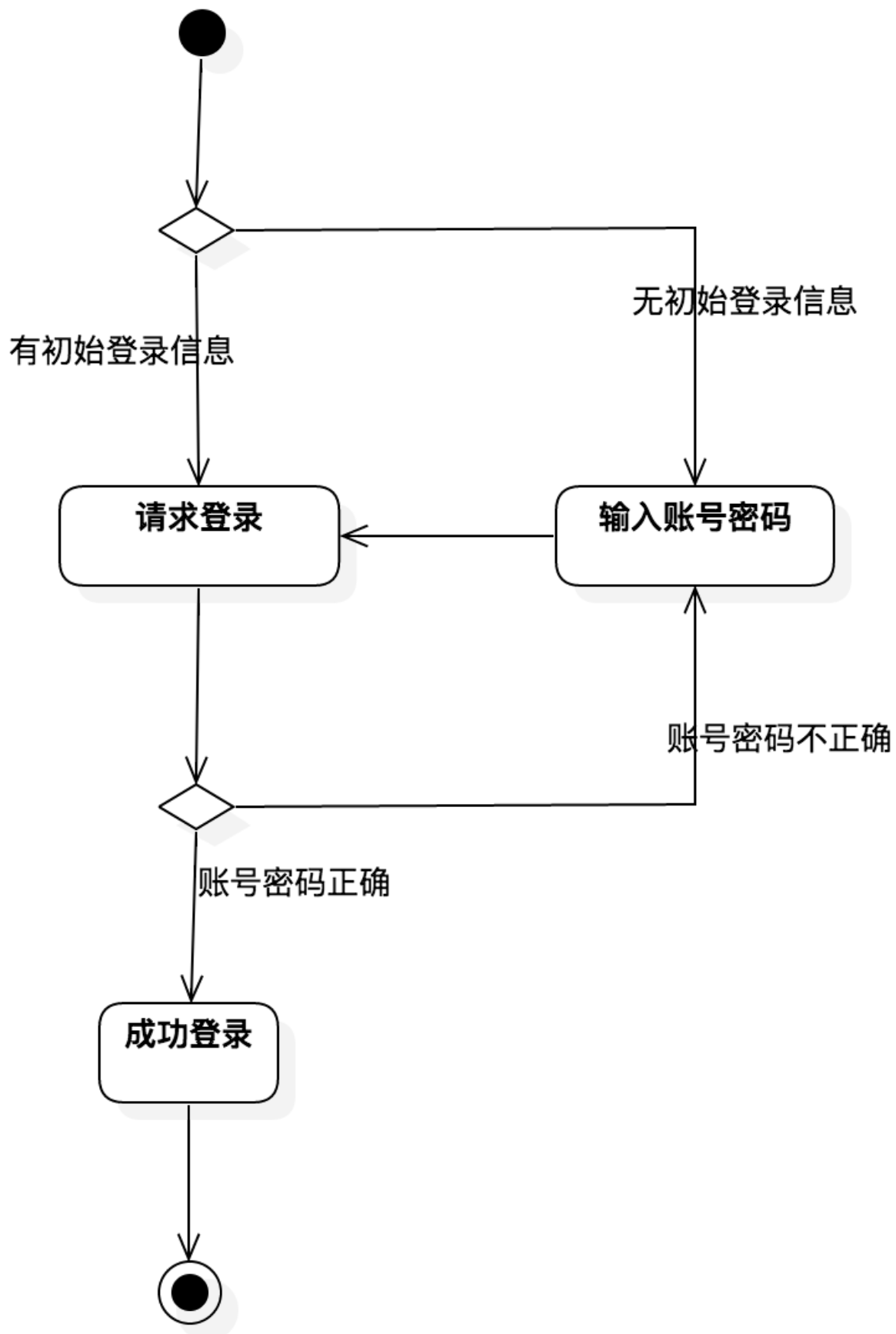
用户可以在衣物管理页面进行衣物管理。用户可以选择不同的衣柜、不同的分类，进行衣物的浏览。用户可以选中几件衣物，点击删除按钮后删除选中的衣物以及所有包含删除的衣物的穿搭。用户选择添加衣物，进入添加衣物页面，用户输入衣物信息后可以添加衣物。用户可以选中几件衣物后将这几件衣物添加为一个新的穿搭。

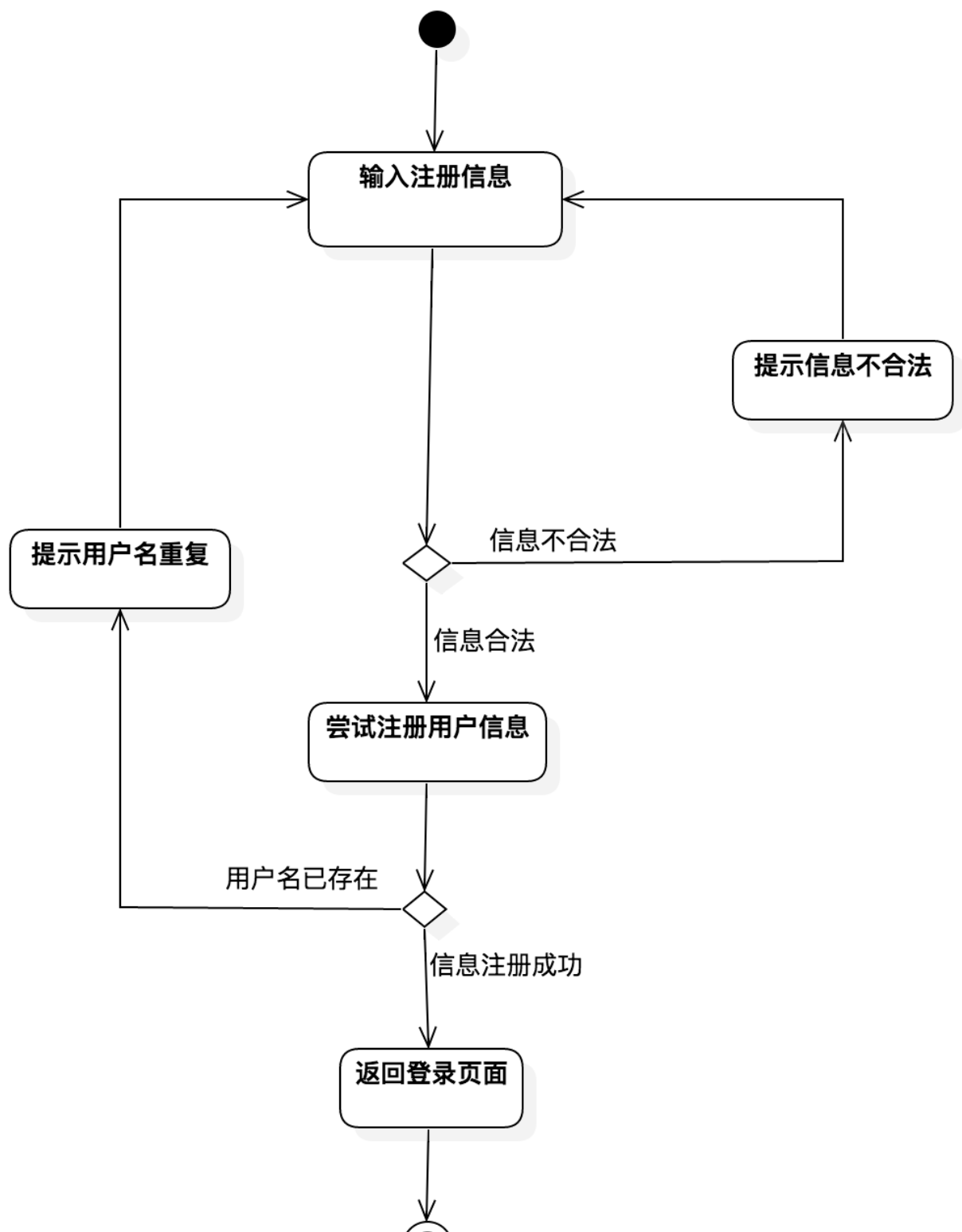
用户可以在衣柜管理页面添加衣柜或者删除衣柜。删除衣柜同时会删除这个衣柜内的全部衣物。

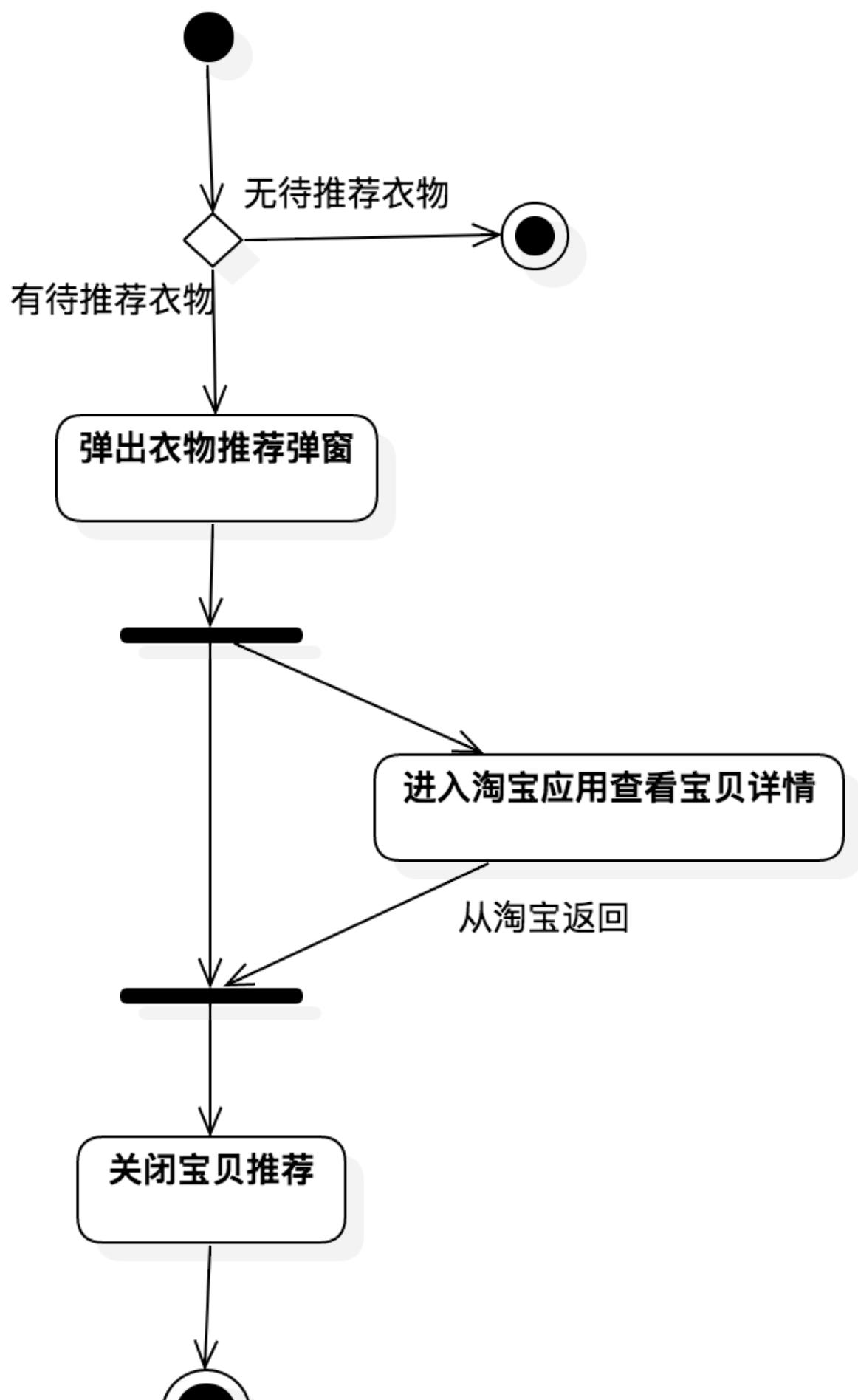
2.4需求分析

2.4.1 功能模型—活动图

2.4.1.1登录

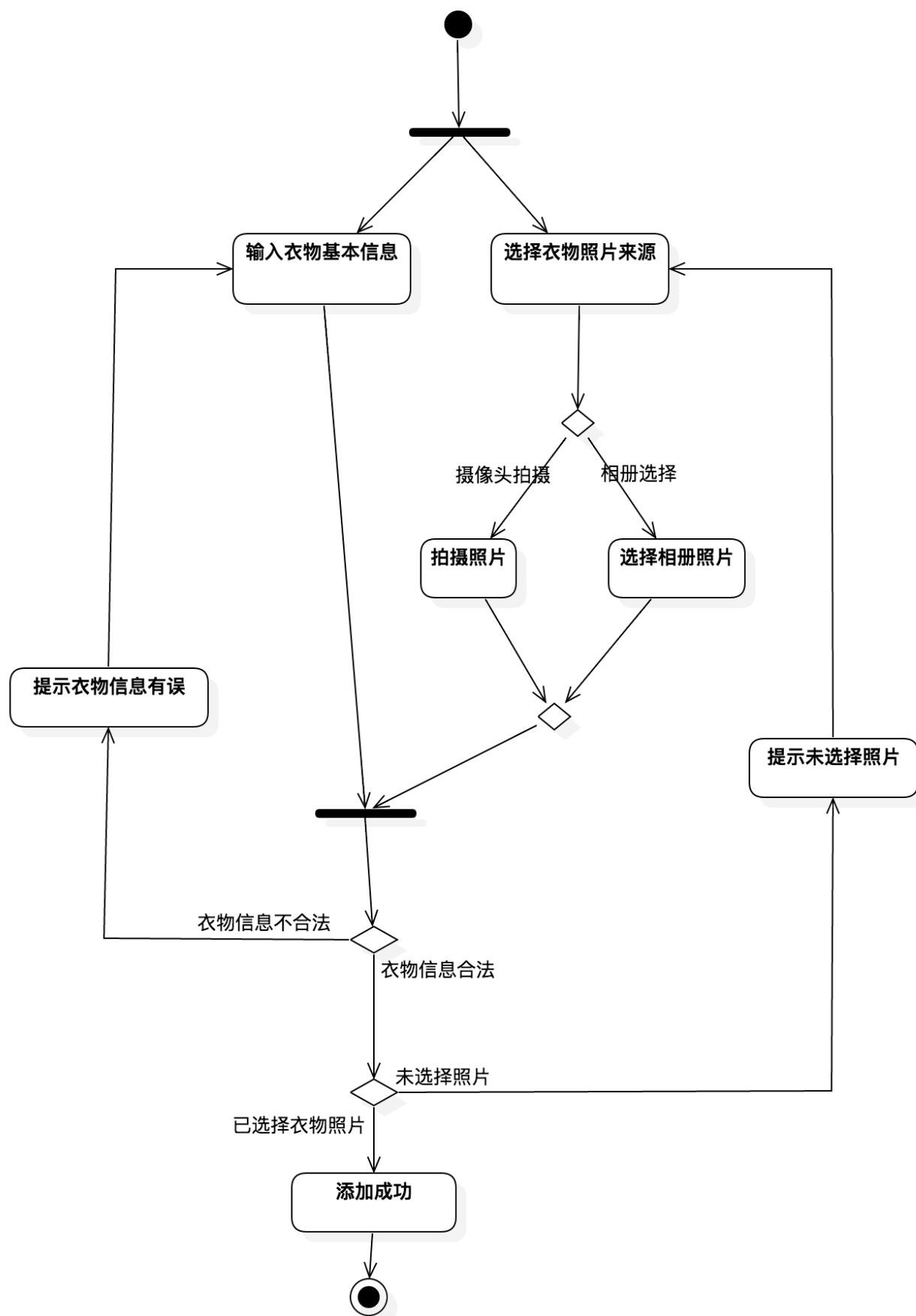


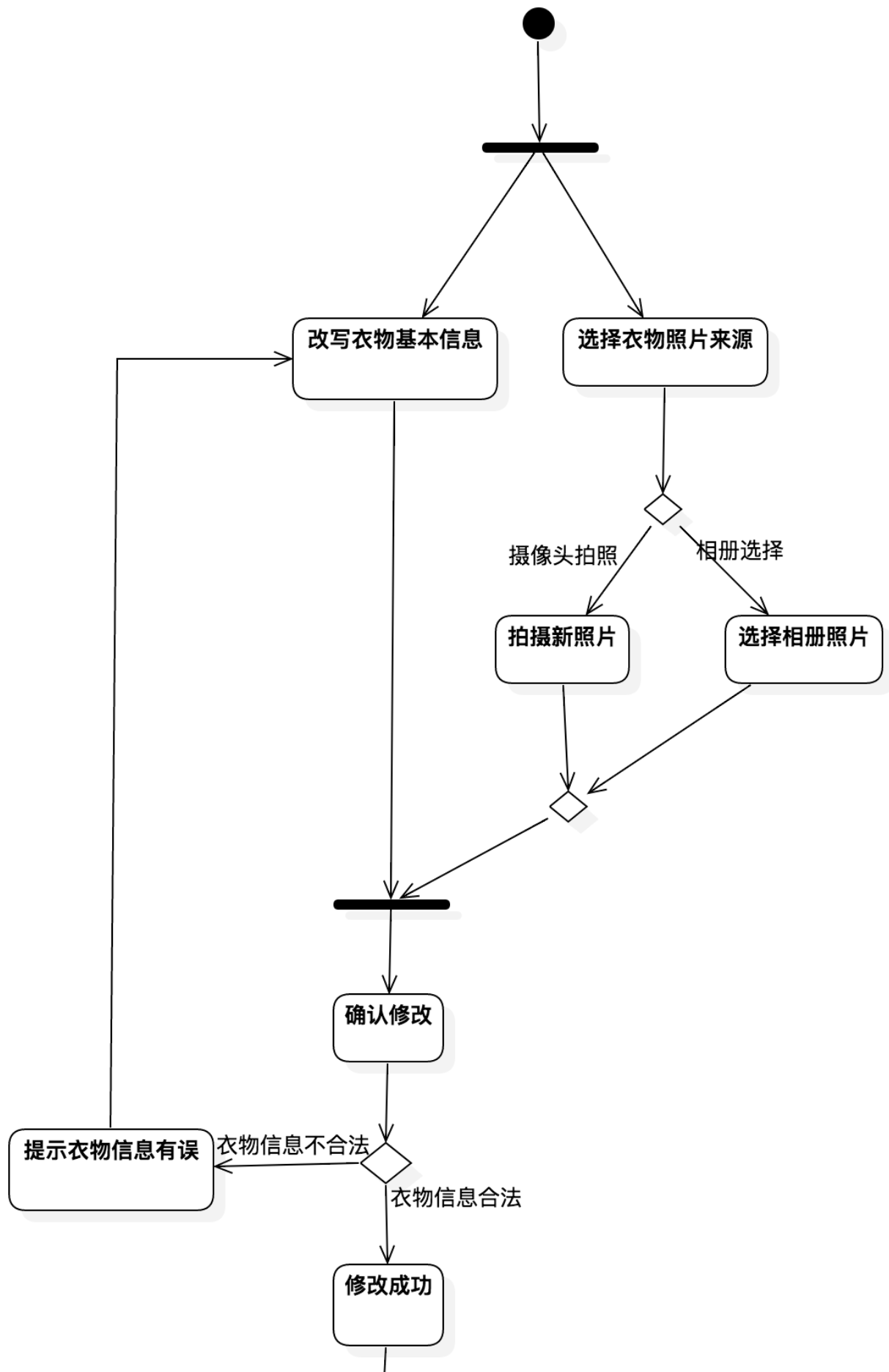






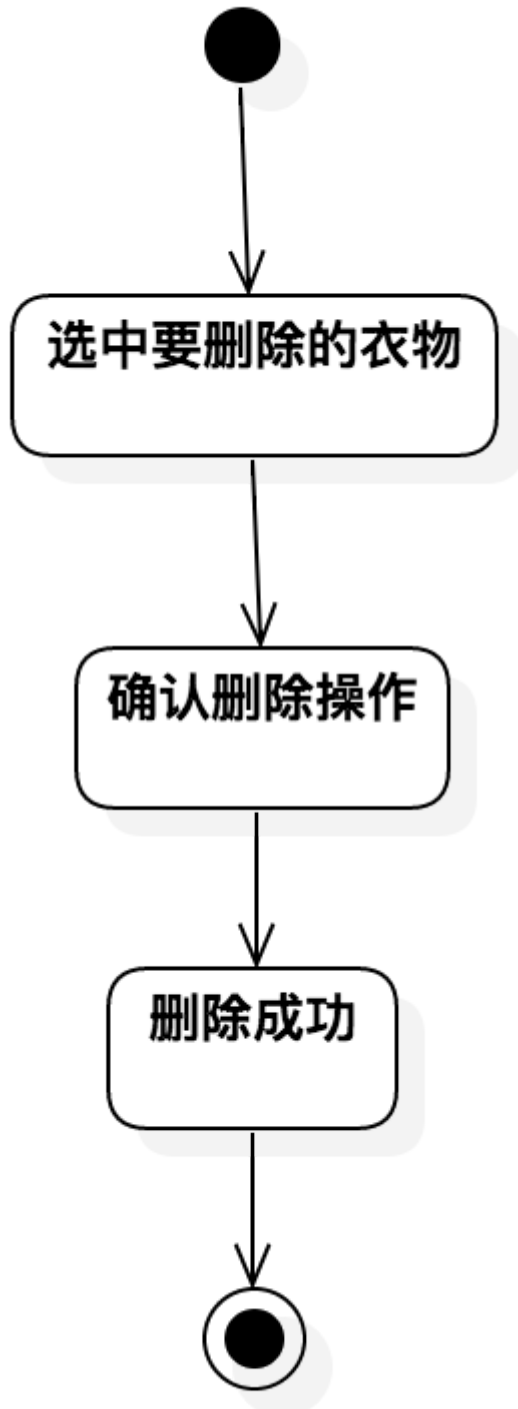
2.4.1.4添加衣物



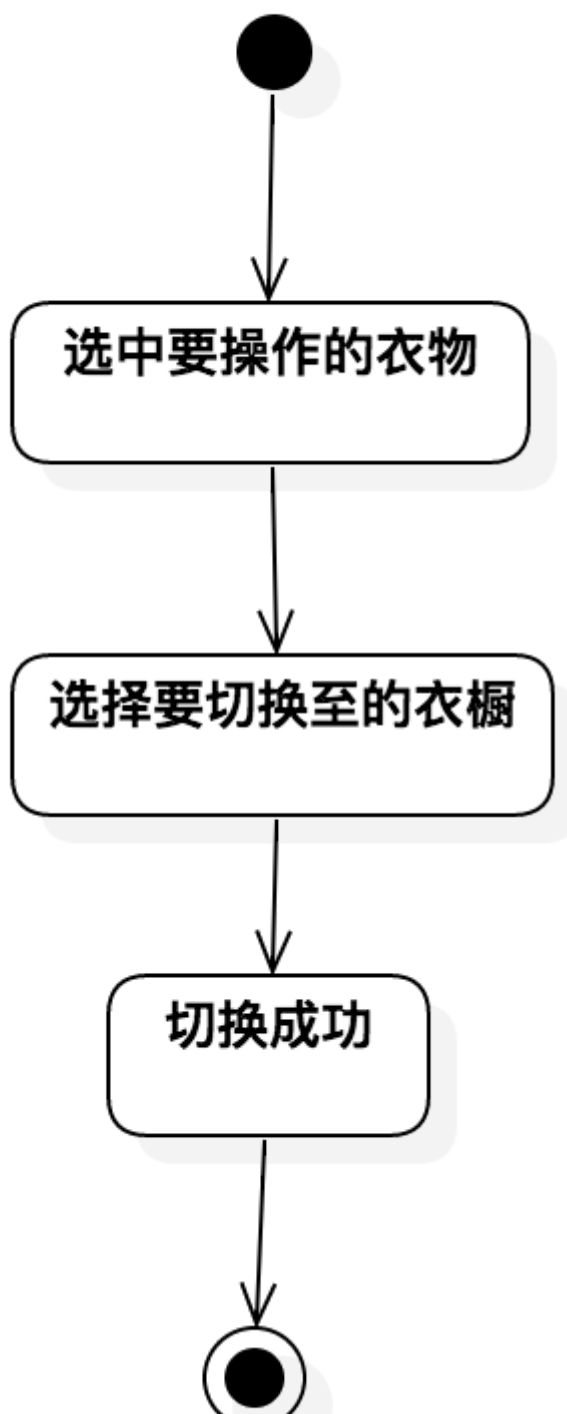




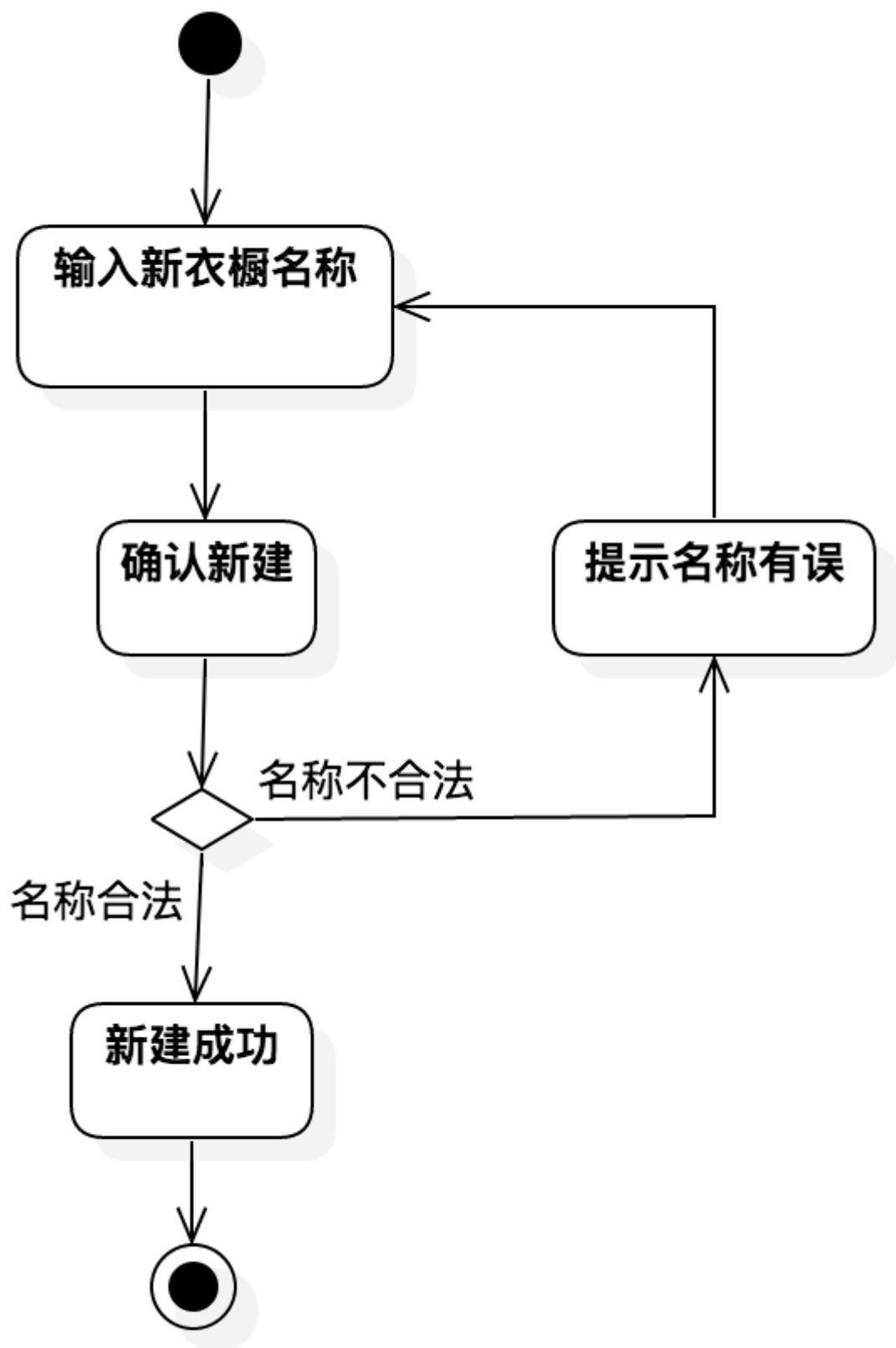
2.4.1.6删除衣物

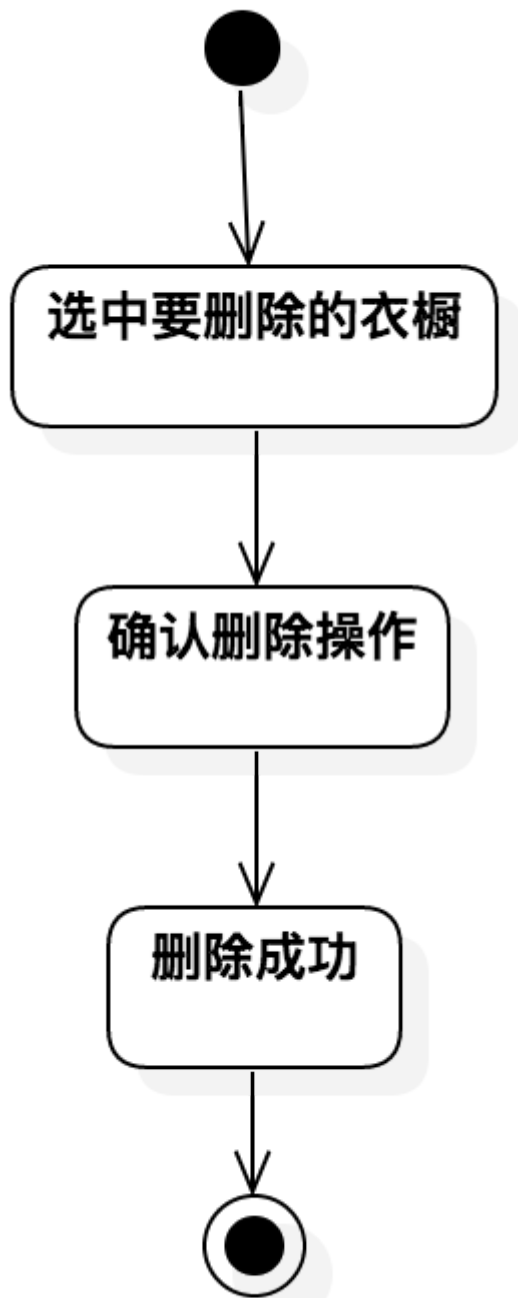


2.4.1.7衣物切换衣橱

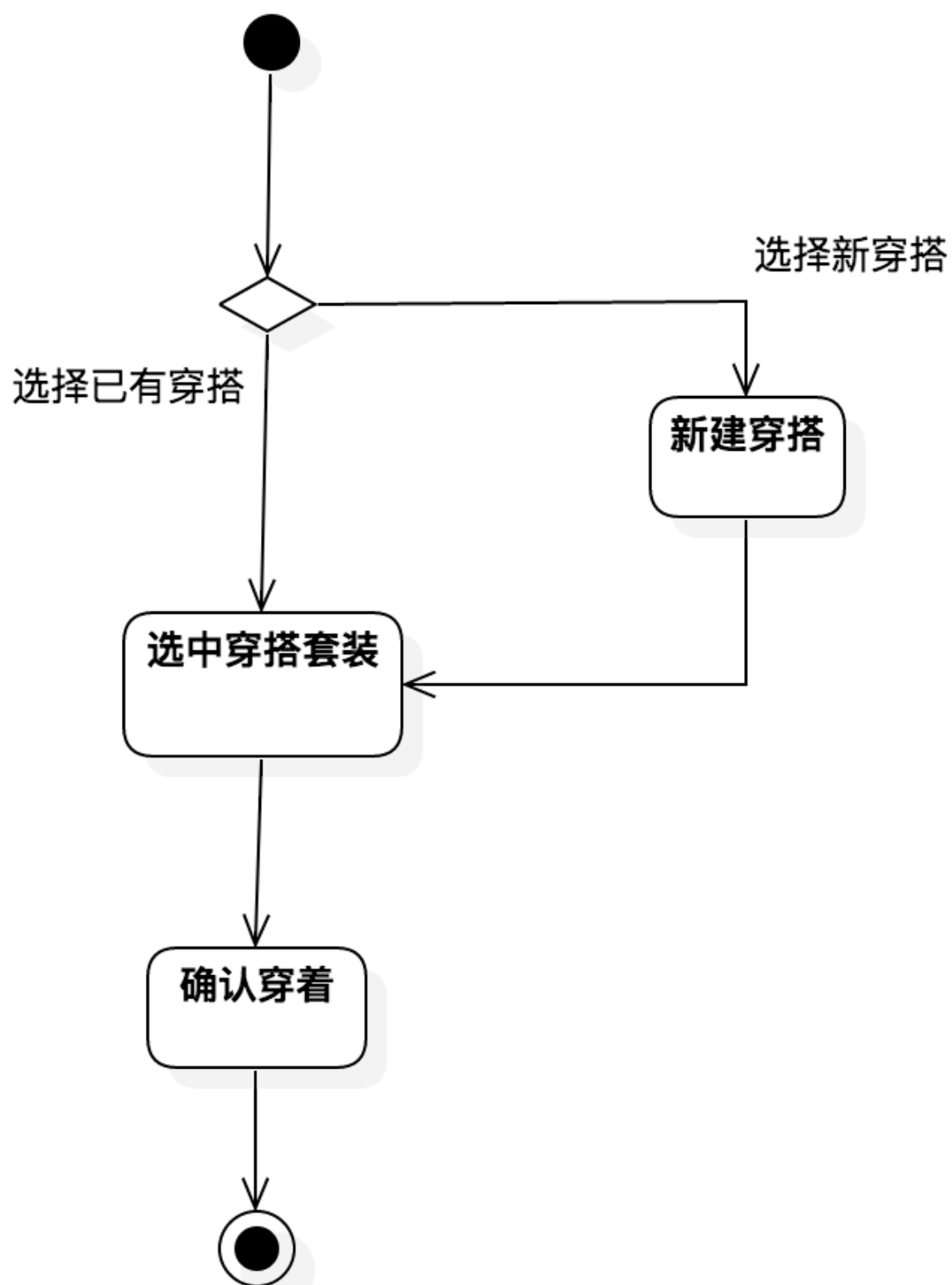


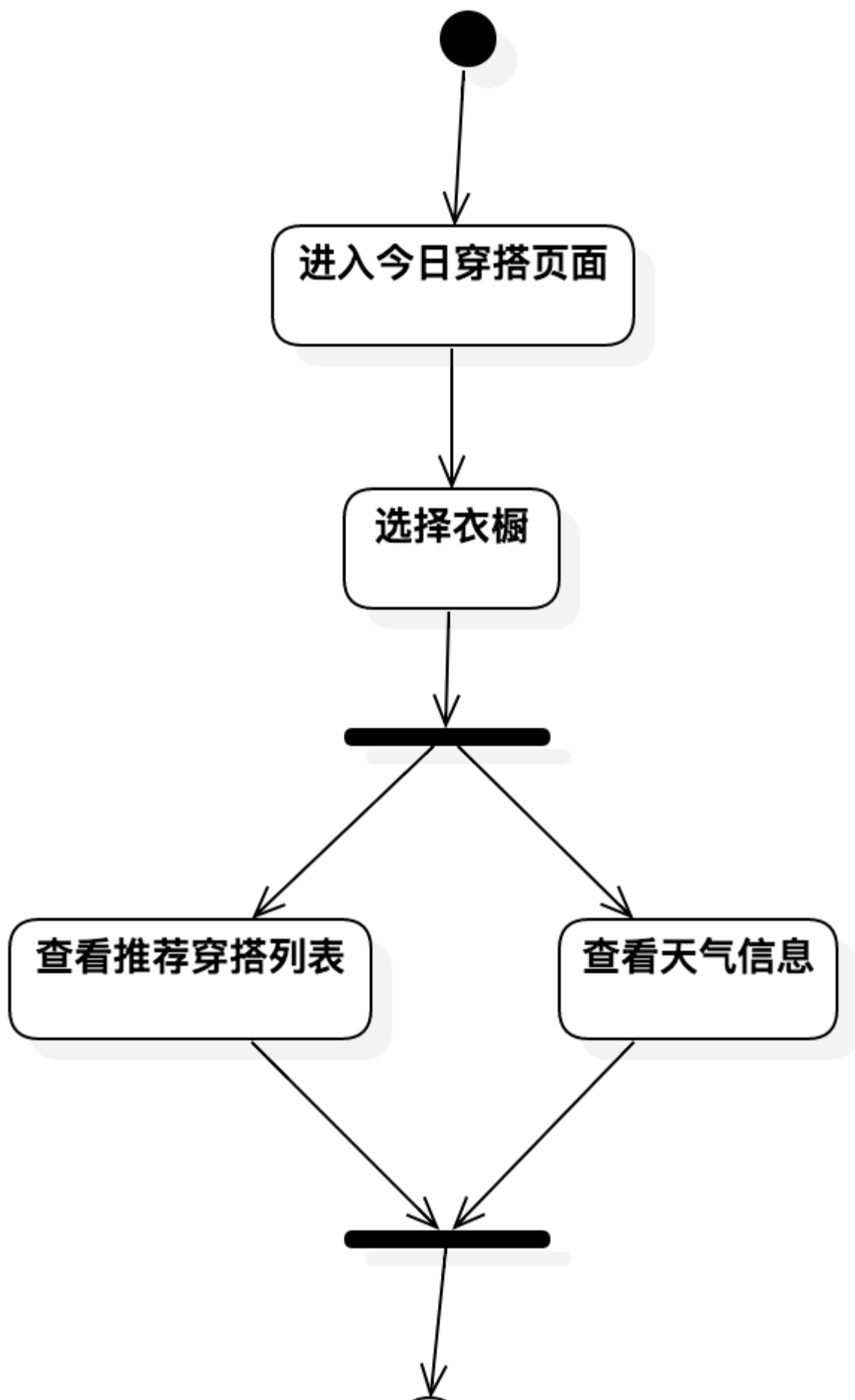
2.4.1.8新建衣橱





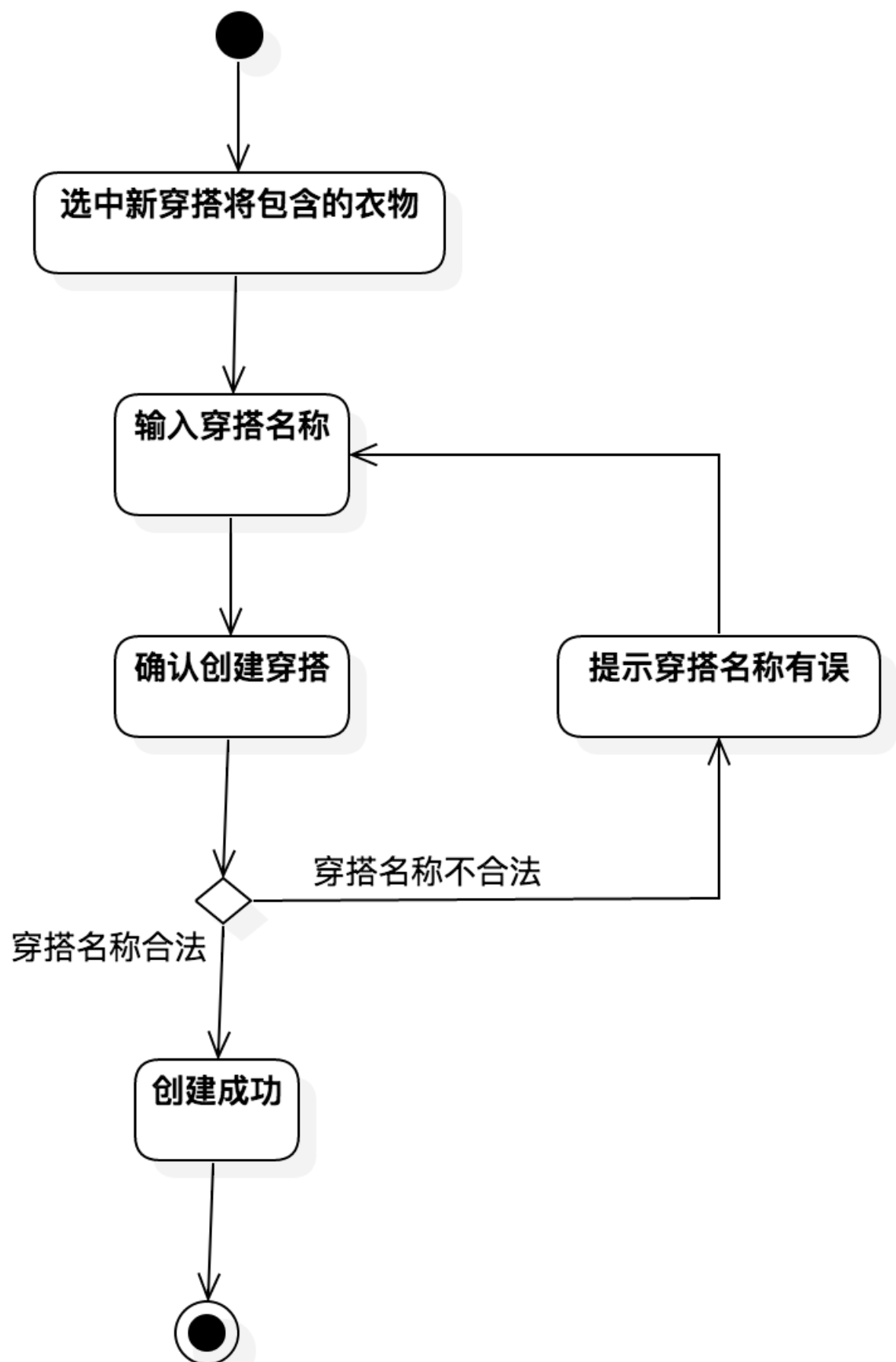
2.4.1.10选择今日穿搭



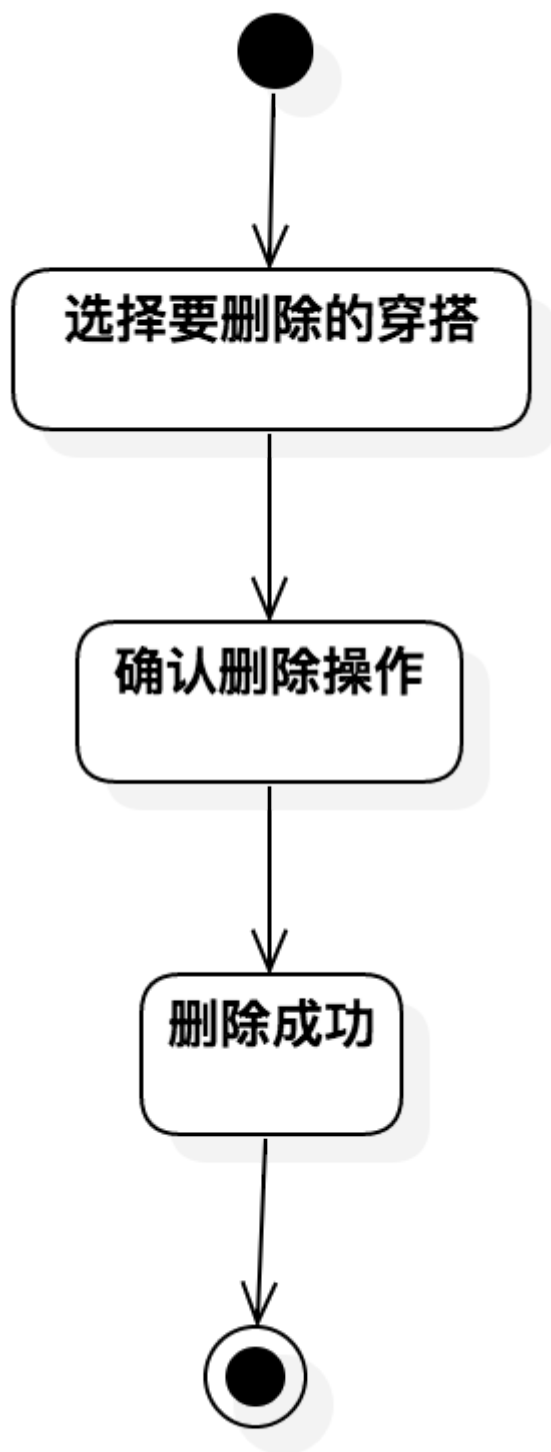




2.4.1.12新建穿搭套装



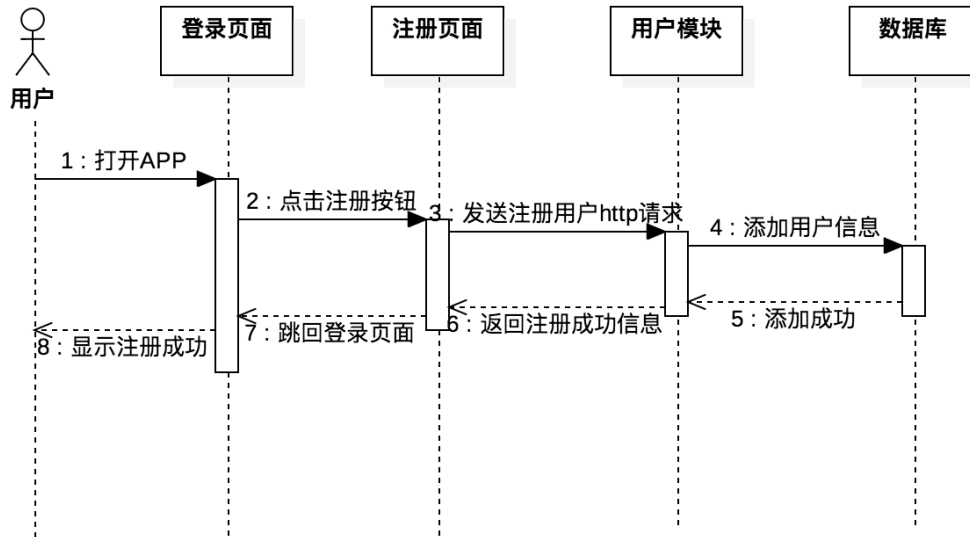
2.4.1.13删除穿搭套装



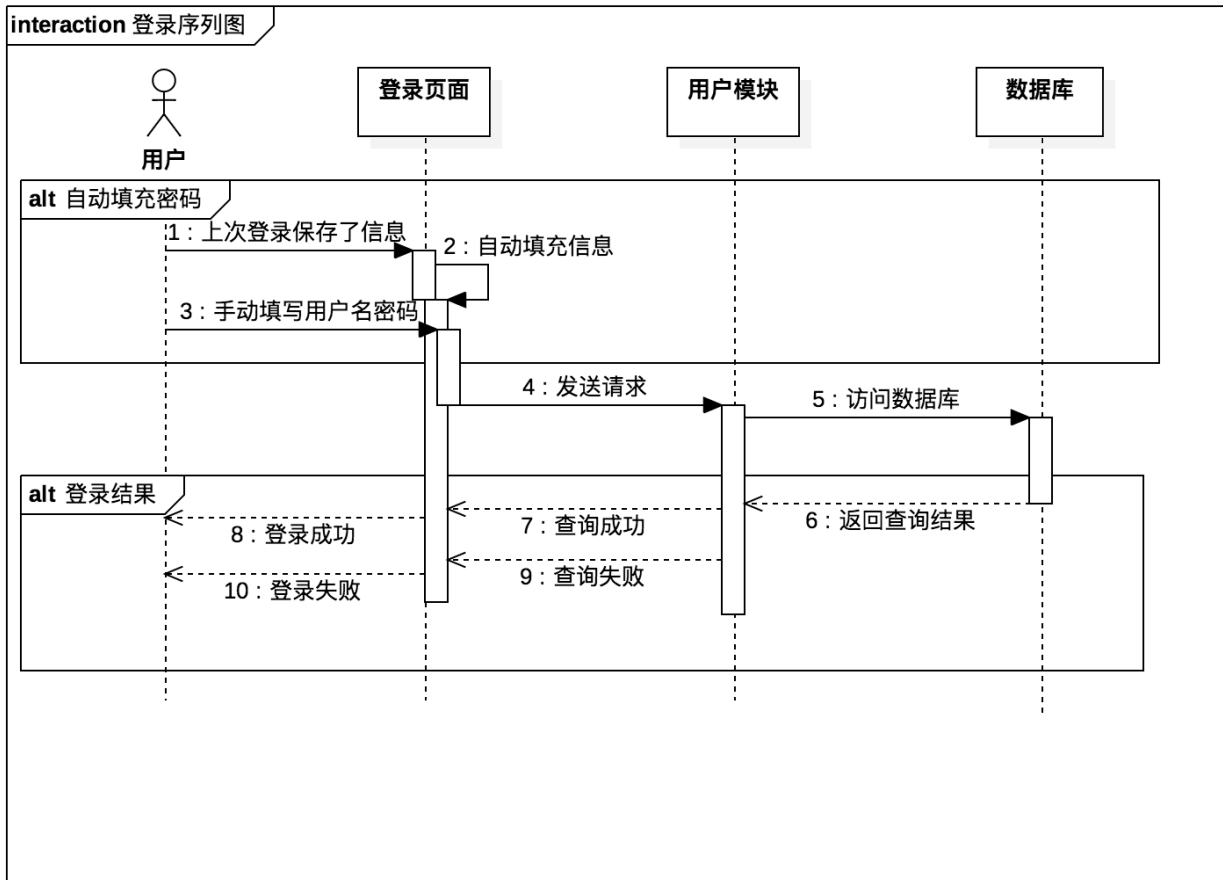
2.4.2数据模型--分析类图

[illegible]

interaction 注册序列图

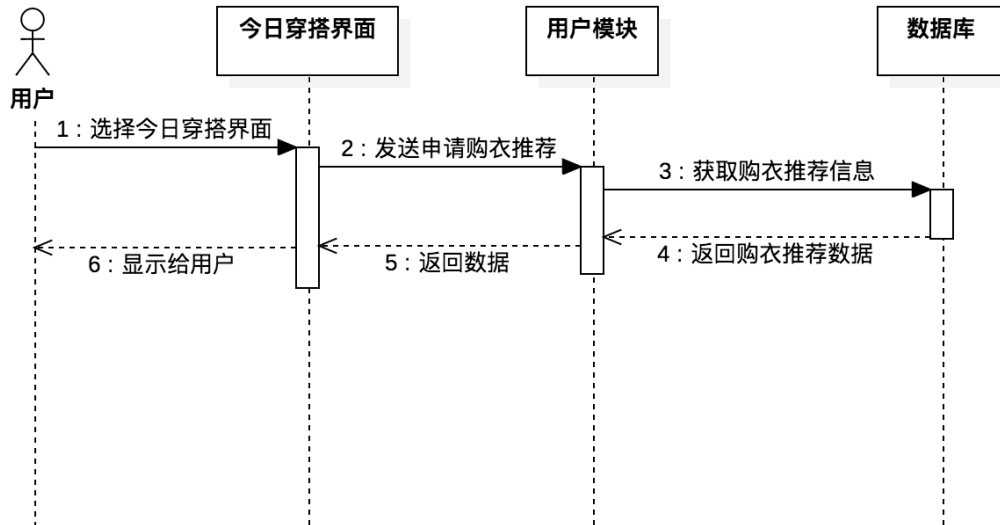


2.4.3.3登录

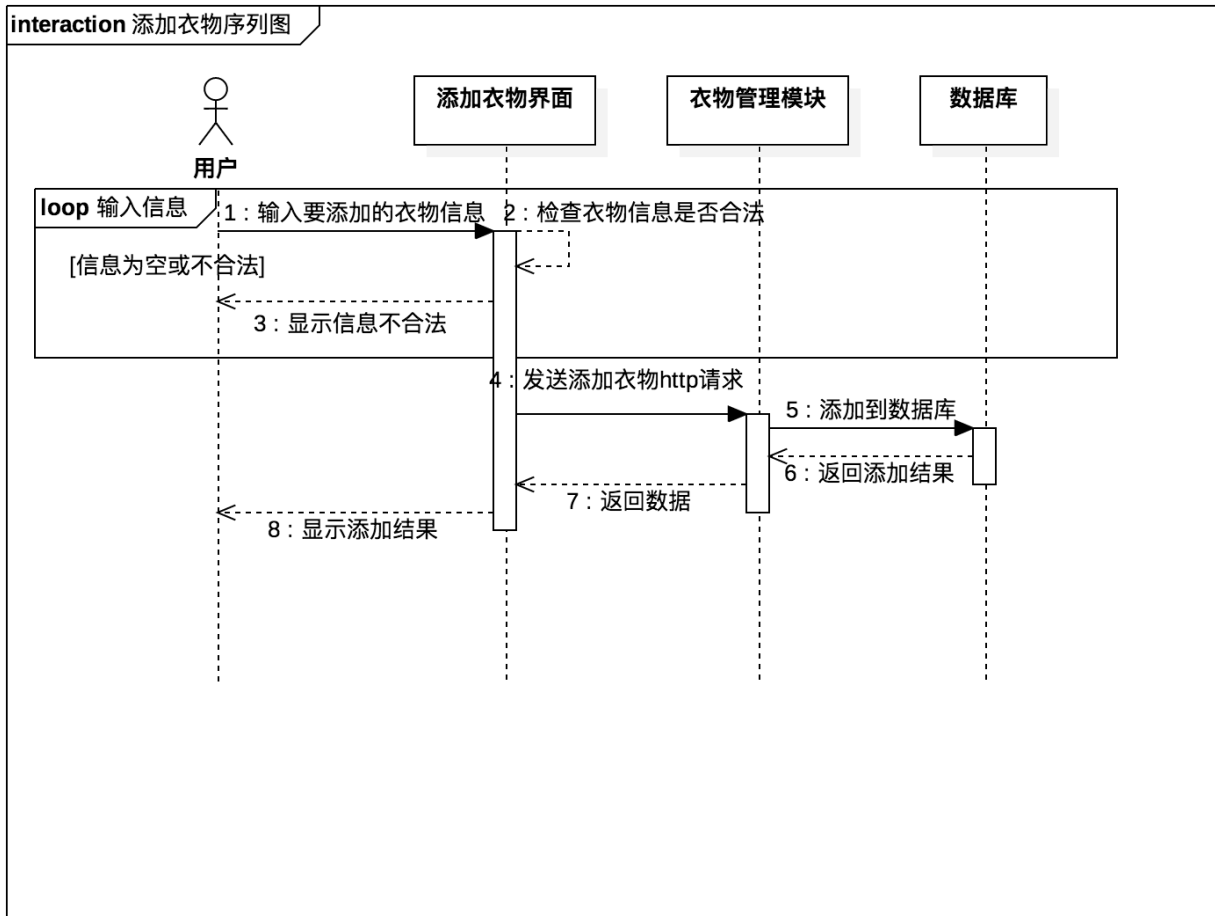


2.4.3.4获取购衣推荐

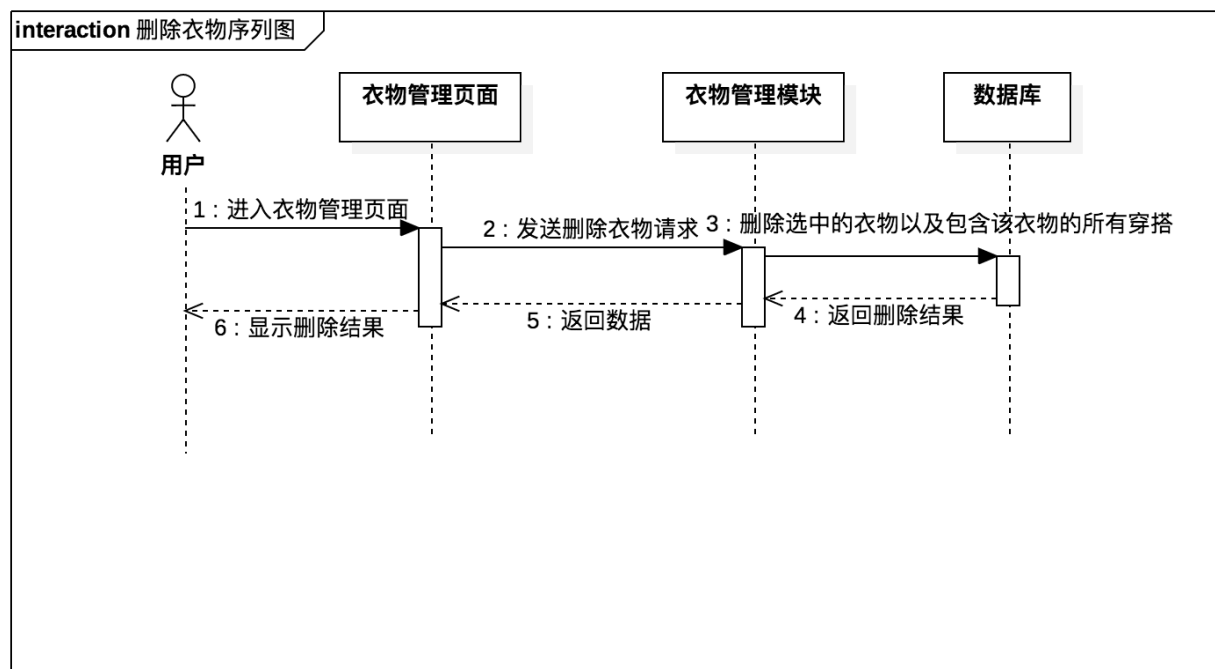
interaction 获取购衣推荐序列图



2.4.3.5添加衣物

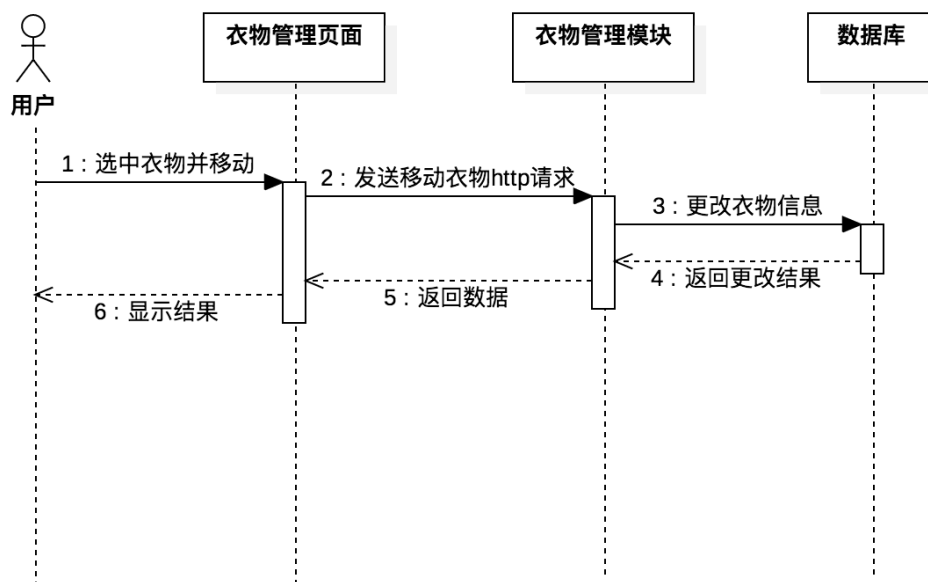


2.4.3.6删除衣物

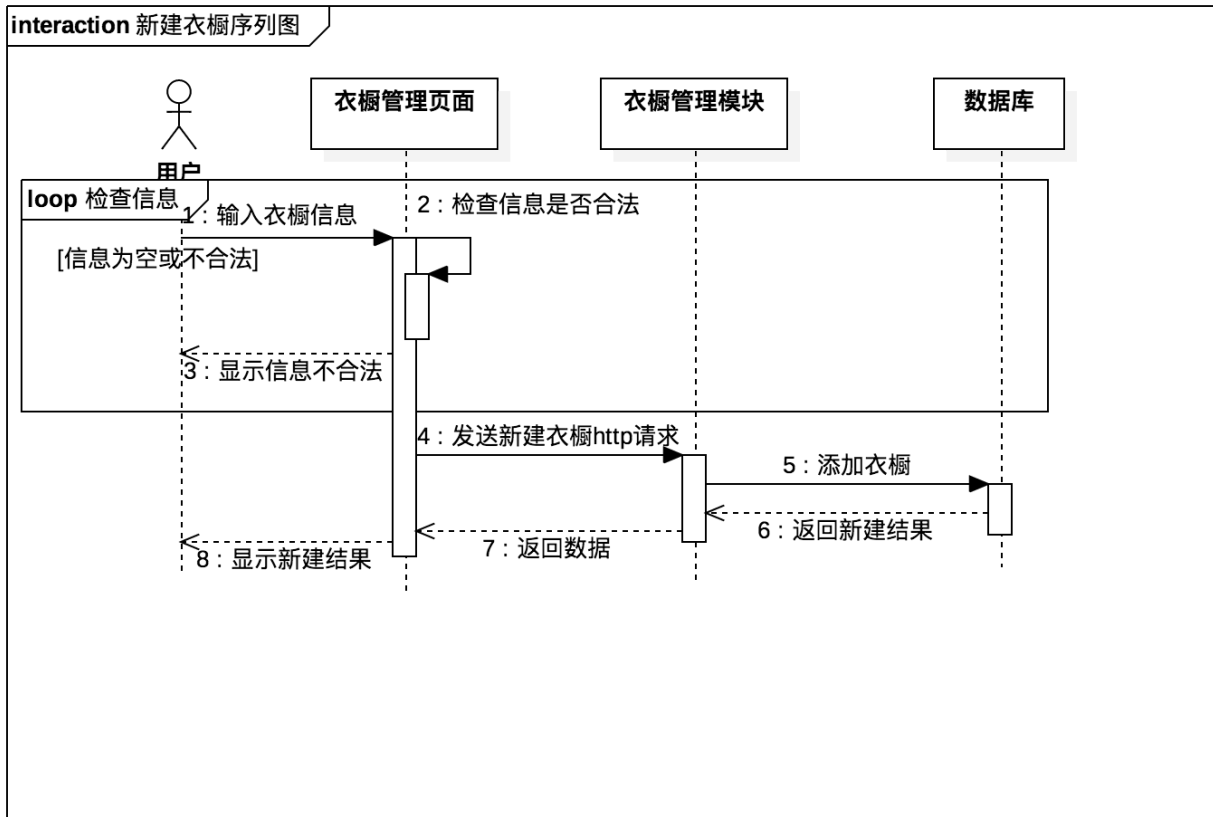


2.4.3.7衣物切换衣橱

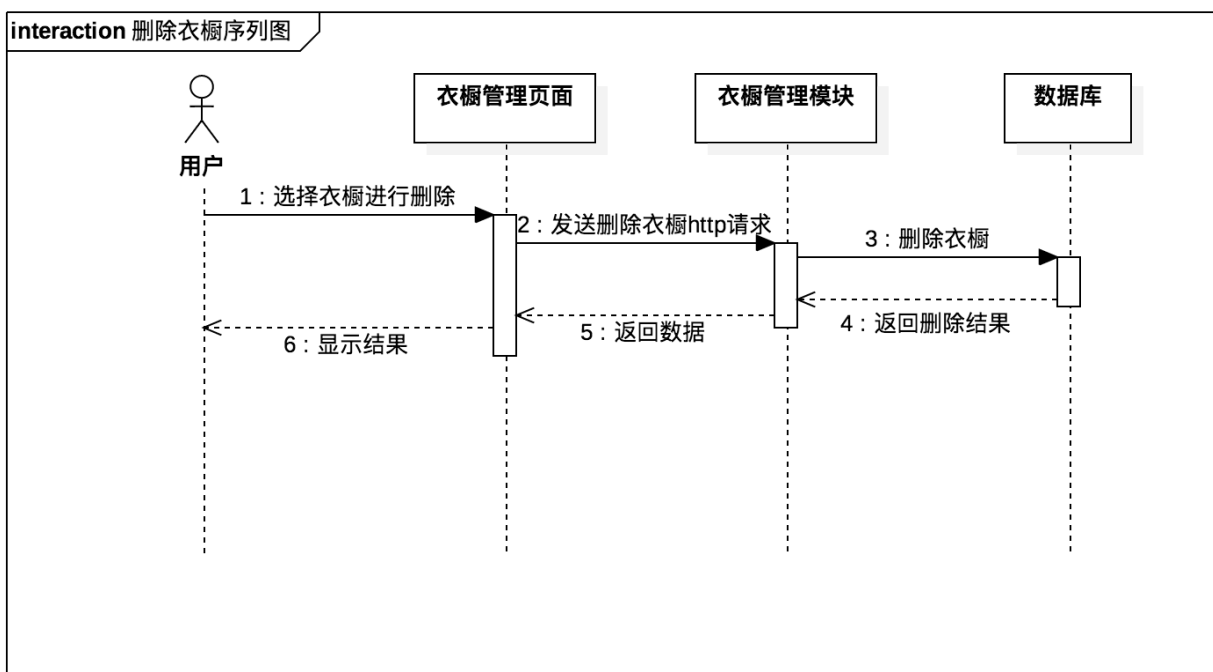
interaction 衣物切换衣橱序列图



2.4.3.8新建衣橱

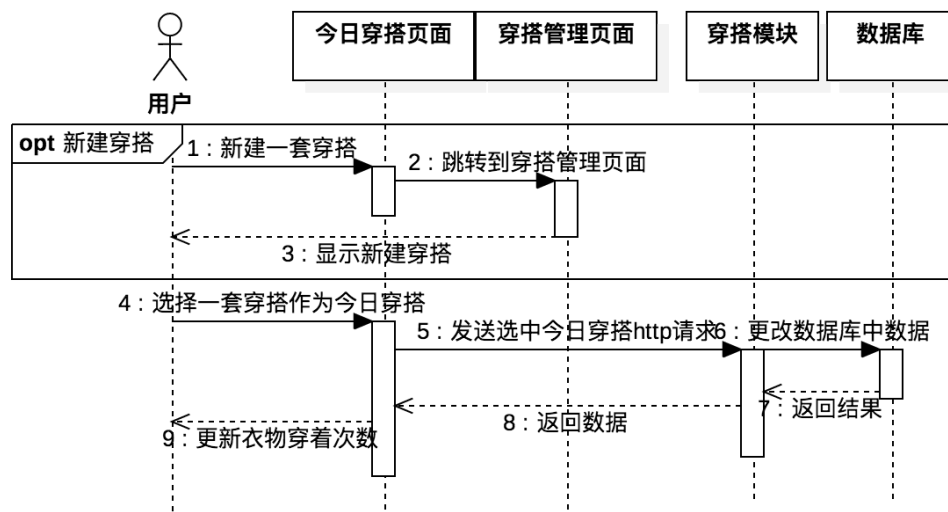


2.4.3.9删除衣橱



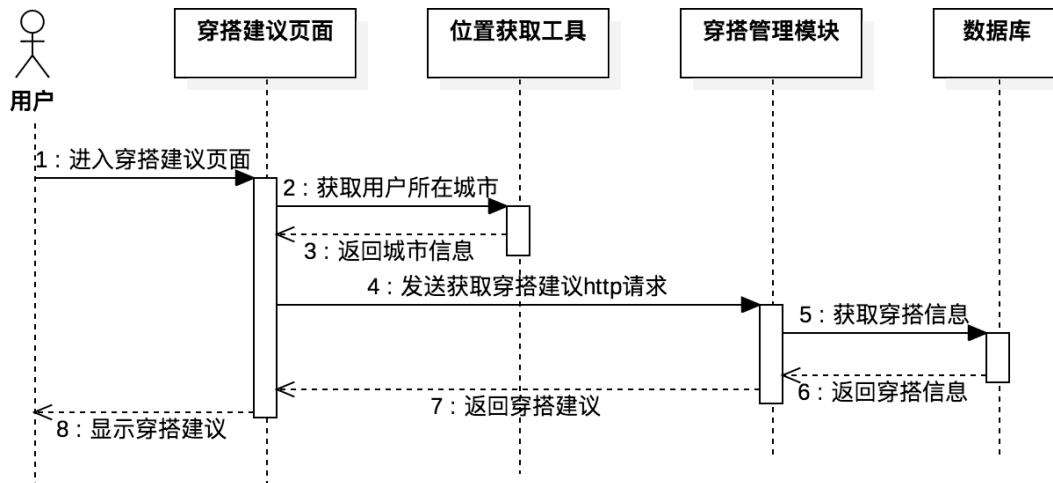
2.4.3.10选择今日穿搭

interaction 选择今日穿搭序列图

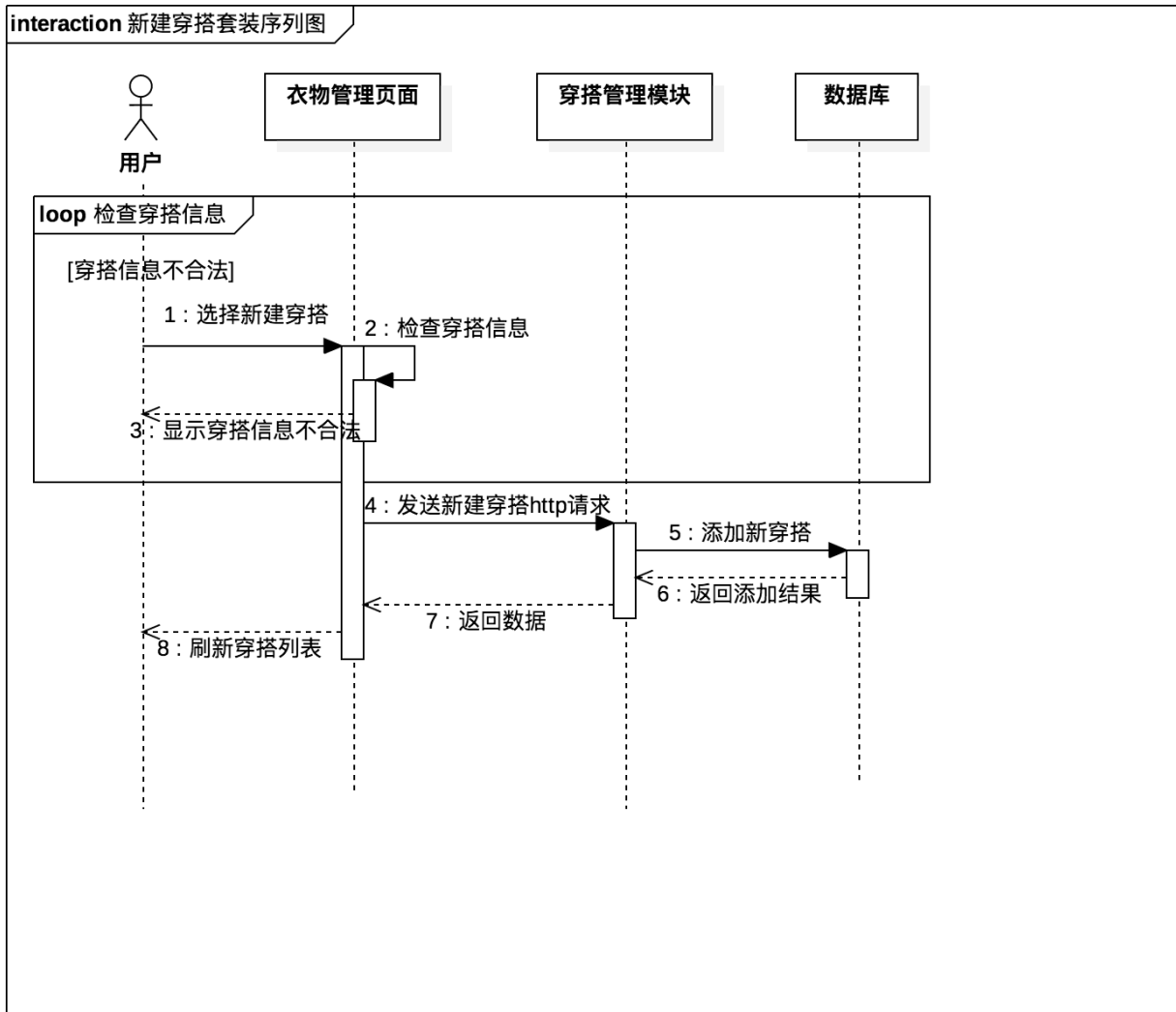


2.4.3.11获取穿搭建议

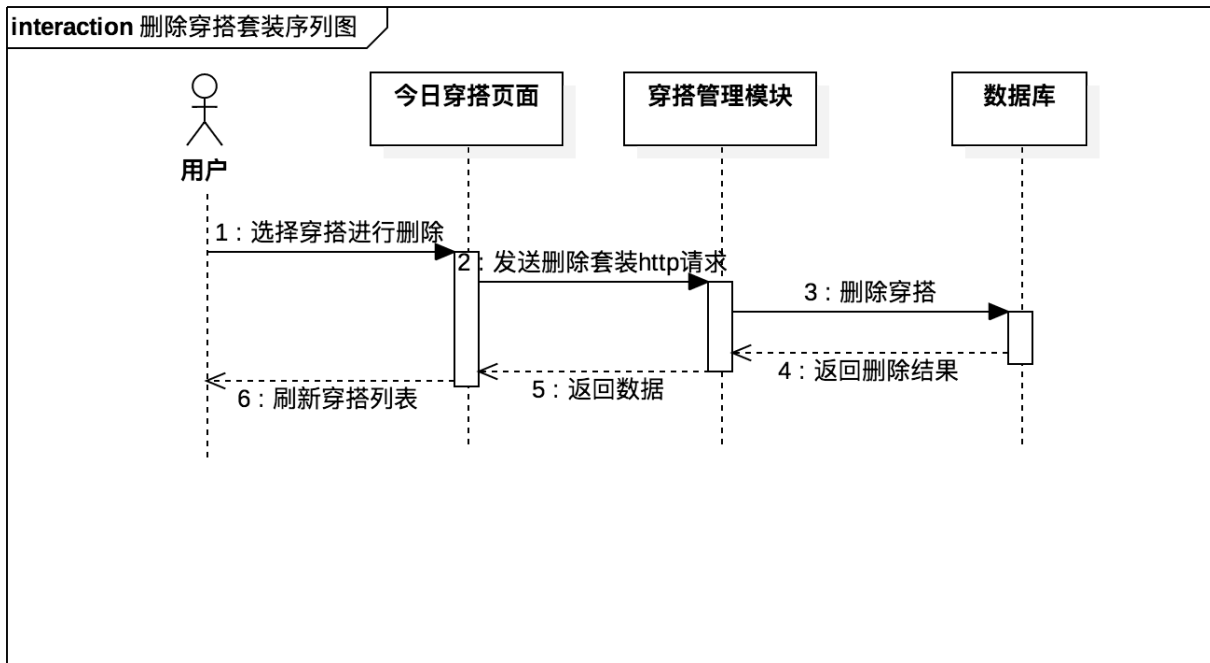
interaction 获取穿搭建议序列图



2.4.3.12新建穿搭套装



2.4.3.13删除穿搭套装



三 非功能需求

3.1性能要求

3.1.1精度

天气信息的准确度精确到摄氏度。

3.1.2时间特性要求

响应时间：

- 1) 对于加载APP页面的操作，要求响应时间在 1~2s 之内；
- 2) 用户对APP按钮进行操作之后，响应时间不超过1s；
- 3) 在网络状况良好的情况下，对于衣物列表的的加载，响应时间不应超过3s；
- 4) 对于衣物、衣橱和穿搭的增删改操作，响应时间应不超过1s；
- 5) 涉及与第三方平台(如天气api)交互的部分，响应时间取决于与第三方平台的数据传输速度。

更新处理时间：

数据库的更新涉及网络数据传输，取决于网络传输速度。但正常情况下，数据库更新时间不应超过 1s，网络状况较差的情况下，不应超过 4s。

数据的转换和界面更新时间：

界面加载采用异步加载的方式，优先加载文字。文字更新时间不应超过0.5s，图片更新时间不应超过3s。

3.1.3输入输出要求

要求用户上传的衣物和穿搭的照片尺寸300*300以上。

3.2数据管理能力要求

3.2.1常量约定

- 1) 在主页面内，用户接收到的推荐穿搭的个数上限是5个。
- 2) 在衣物分类方面，设定了外套、帽子、内衫、裤裙、袜子、鞋子6种衣物分类。
- 3) 用户上传的衣物图片和穿搭图片的图片压缩比率25%。
- 4) 根据用户喜好进行推荐时需要匹配相似度高的用户，用户匹配数上限为5个。

3.3安全及保密性要求

软件必须严格按照设定的安全机制运行，并有效防止非授权用户进入本系统。软件必须提供对系统中各种码表的维护、补充操作。软件必须按照需求规定记录各种日志。软件对用户的所有错误操作或不合法操作进行检查，并给出提示信息。用户必须对系统中的材料成本信息进行维护。mysql自身提供了对数据的安全保护措施，数据进行集中管理同时避免了数据的不一致和冗余，在mysql的版本中，只有管理中心数据库服务器的成员持有对数据访问的权限，

3.4灵活性要求

采用敏捷开发SCRUM模型，灵活接受变化，团队以两周为一个sprint，定期开ScrumDaily会议，并进行Retrospective 总结会议。依照项目的实际需求和之前sprint的情况，制定下一个 sprint 的工作计划。整个团队采取自组织的工作模式，所有成员均担当多方面任务，并根据个人能力主动选择 sprint backlog 中的几个条目进行工作，无需上级干预。

采用Git版本控制系统进行版本控制。

软件架构采用MVC模式，使得界面和逻辑充分脱耦，高灵活度。

在类的层面实现高程度的内聚和封装。

日后若信息量较大，则系统可相应增加服务器实现扩展。后端可移植性较强，重新编译即可运行在Windows、Linux、MaxOS系统上。当操作方式，运行环境，开发计划等发生变化时，只需对数据库本身的文件和记录做处理便可满足需求。

3.5其他专门要求

3.5.1 可维护性

可维护性是指在不影响系统其他部分的情况下修改现有系统功能中问题或缺陷的能力。

身为开发人员，我们在创建和设计系统架构时，为了提高系统的可维护性，必须考虑以下几个方面的要素:低耦合、高内聚合系统文档记录。本系统将采用严格的软件工程的规范进行开发，并采用良好的设计模式保证系统各模块之间的低耦合及模块之间的高内聚。

本系统的所有代码将会被注释，对于系统所有代码，我们会生成详尽的技术文档。对于系统开发过程可能出现的报错，我们将以文档的方式详细罗列报错码及对应的报错信息。

3.5.2 可扩展性

系统建成后，应在现行系统上不需要做大的改动或不影响整个系统结构，就可以增加功能模块，这就必须在系统设计时留有接口，提高可扩展性和维护性，这样就方便在 后期的维护过程中根据用户的需求添加相应的功能，同时不会影响系统其他功能模块的 正常运行。除此之外，还应当保证所有文档说明的详细、完整度，使其他接收者或新加入开发者能够快速地了解APP的情况，并且正确地在原有基础上进行更新和加强。

3.5.3 灾难恢复

由于本应用为服务器-客户端模式的移动应用，用户数据基本全都保存在服务器端，因此服务端应当负责数据备份并保证数据的安全性。为了防止网络方面的攻击，本系统应当具备防治黑客入侵随意篡改用户已经生成好的线路的技术，尤其是系统内的一些核心数据，应采取数据加密存储技术，防止不法分子盗用数据，造成用户数据泄露。即使发生灾难造成数据丢失，系统立即通过备份数据进行恢复，并且重新备份，保证系统中始终保持至少一份数据备份。

四 运行环境规定

4.1设备

服务器：CPU六核，十二线程，16G内存，带宽50Mbps

客户端：1.2GHz主频处理器和1GBRAM

4.2支持软件

1) UI设计：material design （tools：Sketch Zeplin）

2) API测试：Postman

3) 客户端数据存储：SharedPreferences

使用位置：登录页面内用于存储用户上次登录的id和密码

4) 传感器：Accelerometer Sensor摇晃传感器 Vibrate震动反馈

使用位置：Main Activity主页上推荐穿搭后用户可以摇晃手机得到一个随机的推荐

5) 图片存储方法：存在http请求body中的form-data中，发送至服务器存储

6) 网络：Retrofit和OkHttp3

7) 代码托管平台:GitHub

8) 获取天气信息

9) 逆地理编码，使用位置：获取用户所在城市，根据城市获取天气信息

4.3接口

1)天气查询接口

通过提供用户所在位置获取实时的天气信息，包括温度、风力等基本天气信息要素。

2)逆地理编码接口

通过提供的经纬度坐标信息获取对应位置的政治区域名称。

3)电商（淘宝）搜索接口

通过已有的衣物图片找到相似的淘宝宝贝。

4)安卓系统相册摄像头接口

调用系统的相册或摄像头获取照片。

5)后台数据库接口

通过连接数据库实现数据的持久化存储和加载。

五 需求跟踪

见ScrumDocument文件夹中的ProjectStoryBacklog.xls，SprintBacklog.xls，ProjectEverySprintEstimation.xls文件。