

2024-10-23

Software Requirements Specification

软件需求规约

《济事通》同济学生实用平台项目

2250736 王天阔 2251918 刘骏伟 2150021 倪佳瑞

目 录

1. 简介	1
2. 用例建模	2
2.1 功能性需求与顶层建模	2
2.2 交易中介模块	3
2.3 拼单中介模块	7
2.4 信息查阅模块	12
2.5 敏捷需求表	16
3. 术语表	17
4. 补充规约	18
4.1 用户类型	18
4.2 环境描述	18
4.3 假设、限制与依赖	18
4.4 系统接口	19
4.5 非功能性需求	19
5. 初始快照	21
5.1 主页快照	21
5.2 交易中介模块快照	22
5.3 拼单中介模块快照	23
5.4 信息查阅模块快照	24
贡献列表	25
参考资料	26

1. 简介

当今大学生生活与互联网紧密相连，经过我们的调研，我们发现学生对于便捷、高效、简单、全面的校园平台需求正在日益增加。已有的学生服务软件都有着或多或少的缺陷，例如功能使用不方便、功能不全等问题。因此，我们决定构建一个集各种功能于一体的小型学生实用平台应用。

项目主要面向同济在校学生，以解决他们在校园生活中的需求，学生群体具有庞大的潜在用户量，且目前缺乏一个高便捷性服务系统。相比于通用型平台，本项目可以更好地服务学生需求。

我们目前决定以校园交易拼单、信息查阅等为项目的主营业务。交易拼单等功能包括商品发布、交易管理、安全保障、信息发布、人员匹配等环节。同时整合已有常规功能，如信息查询、选课情报、备忘录等，真正解决学生常见问题，切实提高效率和舒适度。

项目具有较大的发展潜力。一方面其用户群体广大，使用成本低廉，因此很容易吸引用户前来尝试使用；另一方面，学校各系统越来越复杂，新功能也越来越多，学生间交流的需求也会随着时间发展而变化，项目功能发展具有远景可扩展性，可以在各种外部系统发展的同时与时俱进，不断焕发生机活力。本项目一方面培养了成员开发整套系统化的软件程序的能力，包括相互交流沟通的团队协作能力和切实将软件工程和系统分析设计理论应用到实际工程中的专业能力，另一方面，项目成员也是项目目标用户的一部分，该项目能够切实方便项目成员的学习生活。

本文档是项目的软件需求规约，是在进行软件需求分析和初步设计的基础上完成的。本文中，我们将整个系统剖分为交易中介、拼单中介和信息查阅三个功能性模块，并对这三个功能性模块进行需求建模。同时针对系统的其他需求，将在补充规约中予以说明。

2. 用例建模

2.1 功能性需求与顶层建模

功能性需求分析：

可以从系统主动参与者的角度考量。作为最主要的参与者，使用系统的学生的需求尤为重要。经过对学生群体的调研和分析，可以得出如下的学生对系统的功能性需求：

交易中介部分：

需求1：新增交易请求。作为交易提出者时，学生需要能够在平台上提出新的交易请求，以寻求合适的交易对象，供其查阅和响应。

需求2：响应交易请求：作为交易接受者时，学生需要能够在平台上寻求并响应其他用户的交易请求。

需求3：管理交易请求：学生应当能够管理自己提出的交易请求，可以根据意愿或请求响应情况进行修改或删除等操作。

需求4：查看交易请求：在响应请求和管理请求时，学生需要能够查看自己提出的或他人提出但尚且无人响应的交易请求。

拼单中介部分：

需求1：新增拼单请求。作为拼单提出者时，学生需要能够在平台上提出新的拼单请求，以寻求合适的拼单对象，供其查阅和响应。

需求2：响应拼单请求：作为拼单接受者时，学生需要能够在平台上寻求并响应其他用户的拼单请求。

需求3：管理拼单请求：学生应当能够管理自己提出的拼单请求，可以根据意愿或请求响应情况进行修改或删除等操作。

需求4：查看拼单请求：在响应请求和管理请求时，学生需要能够查看自己提出的或他人提出但尚且无人响应的拼单请求。

信息查阅部分：

需求1：管理个人备忘录。学生应当能够管理自己的个人备忘录，在系统中完成增删改等操作。

需求2：查阅外部信息。学生应当能够从该系统出发，进入到各个外部系统如教务系统，选课系统，Canvas系统等等，这需要外部系统的参与。

此外，系统管理员也是系统的主动参与者之一。相比于学生而言，管理员对系统的需求要精简很多：

交易中介部分：

需求1：管理交易请求：管理员应当能够管理所有系统中的交易请求，能够对这些请求进行除修改外的操作，例如可以根据某些标准执行关闭请求等操作。

需求2：查看交易请求：在管理请求时，管理者需要能够查看系统中所有的交易请求。

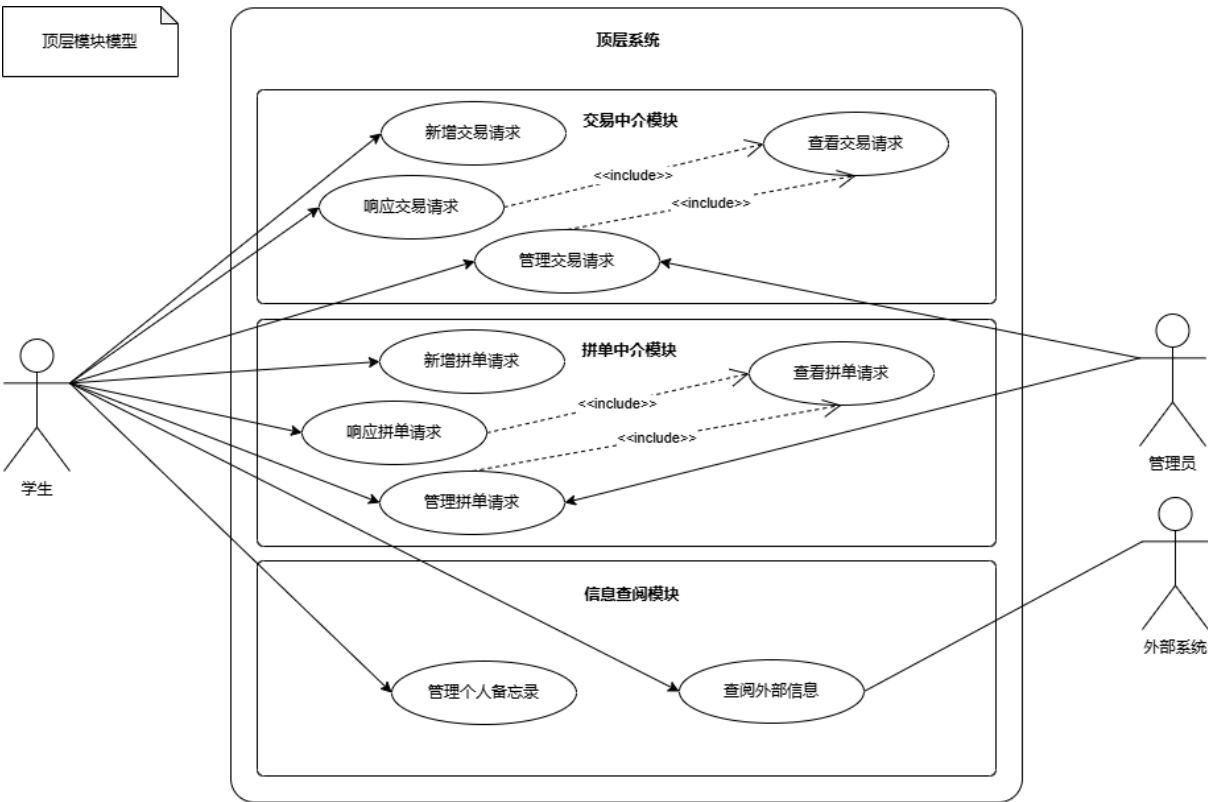
拼单中介部分：

需求1：管理拼单请求：管理员应当能够管理所有系统中的拼单请求，能够对这些请求进行除修改外的操作，例如可以根据某些标准执行关闭请求等操作。

需求2：查看拼单请求：在管理请求时，管理者需要能够查看系统中所有的拼单请求。

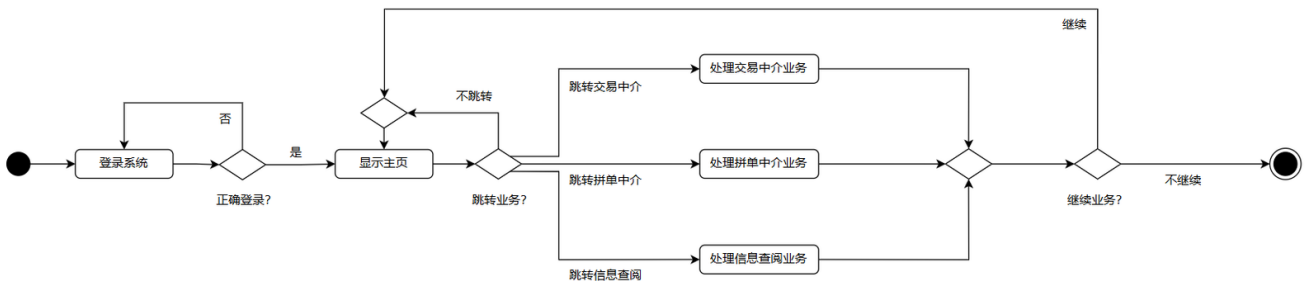
综上，我们可以将系统主动参与者的需求划分为三组，分别是交易部分、拼单部分、信息查阅管理部分。因此，我们按照需求可以将系统划分为三个功能模块，每个功能模块处理一组需求。

顶层用例图：



顶层活动图：

我们将仅展示学生使用本系统时的活动为例。



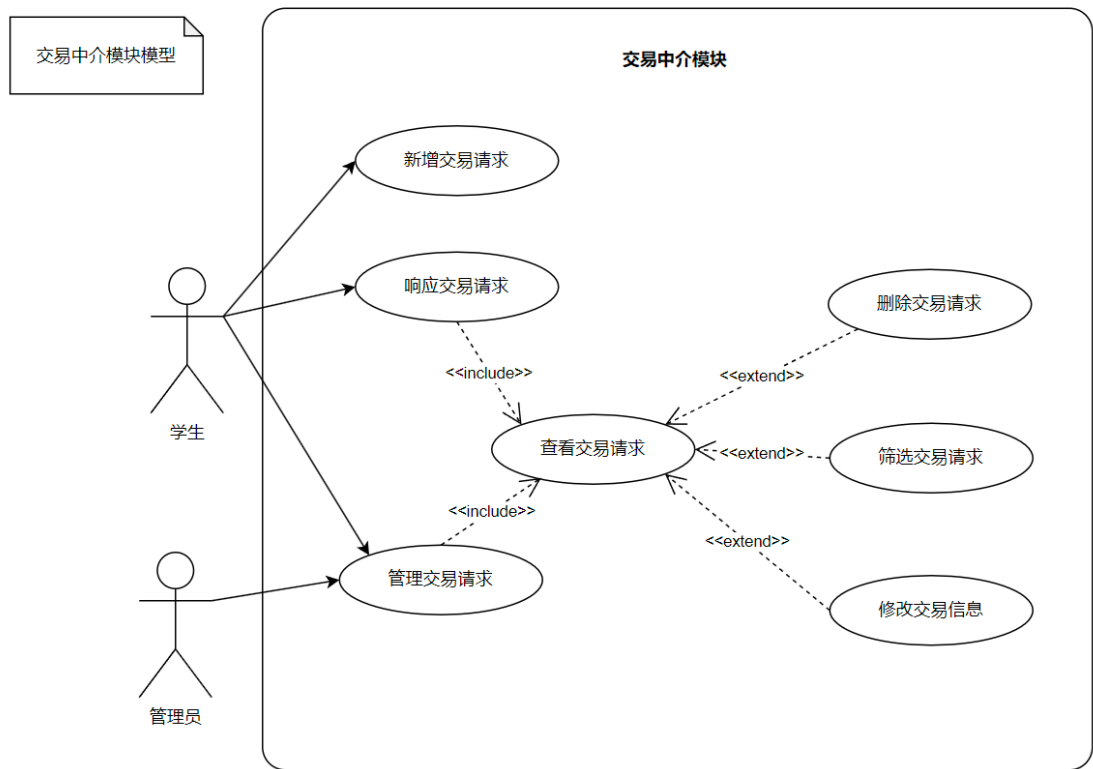
在“处理**业务”活动中，将进入模块的处理逻辑，其具体的活动图描述将在下述模块中给出。

2.2 交易中介模块

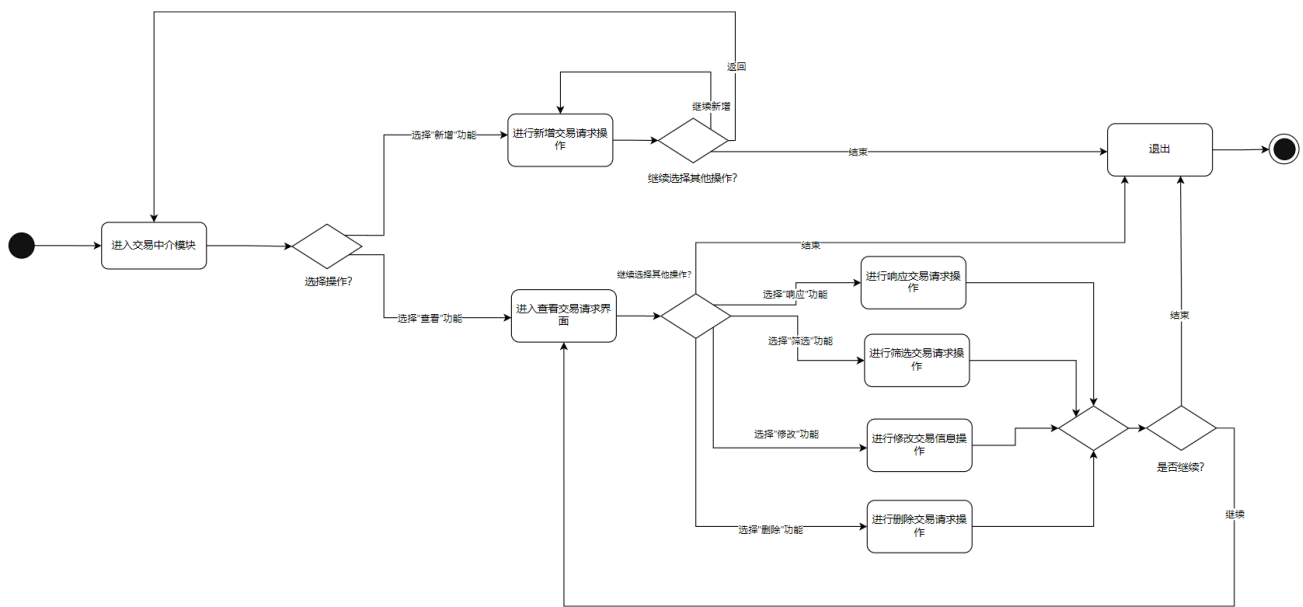
描述：

模块功能：模块处理交易请求的管理和交互，包括新增、响应、查看、删除和修改交易请求。
输入：学生或管理员输入的操作请求，交易请求的具体信息。
输出：交易请求状态更新情况信息与确认信息。
模块边界：模块与用户登录部分和数据库交互，负责学生与管理员的账号数据以及交易请求数据的存取与修改。

模块用例图：



模块活动图：



用例名称：新增交易请求
参与者：学生

前置条件： 学生进入交易中介模块界面
后置操作： 更新新增交易信息
工作流程： <ol style="list-style-type: none"> 1. 学生选择“新增交易请求”选项 2. 学生输入交易请求信息 3. 学生提交交易请求 4. 系统验证输入信息的完整性 5. 若消息完整，系统保存交易请求，返回新增成功消息 6. 若消息不完整，系统提示错误，返回步骤2
扩展流程： 如果步骤1时学生未登录，则进入登录界面，待用户登录后继续后续步骤

用例名称： 响应交易请求
参与者： 学生
前置条件： 学生进入交易中介模块界面，查看交易请求界面
后置操作： 交易请求响应状态更新
工作流程： <ol style="list-style-type: none"> 1. 学生进入交易请求界面 2. 系统根据学生账号信息向数据库查询交易请求的信息 3. 系统返回交易请求的信息列表 4. 学生选择一个需要的交易请求进行响应 5. 学生选择对应的响应操作(同意交易或拒绝交易) 6. 学生提交响应信息 7. 系统接受响应信息 8. 若响应成功，系统返回响应成功消息，并向数据库更新信息 9. 若响应不成功，系统提示错误信息，返回步骤2
扩展流程： 如果步骤1时学生未登录，则进入登录界面，待用户登录后继续后续步骤
活动图： <pre> graph LR Start(()) --> Enter[进入响应交易请求操作] Enter --> SystemReturn[系统返回交易请求列表] SystemReturn --> Select[选择一个进行响应的交易请求] Select --> Choose{选择响应类型?} Choose -- 接受交易 --> SystemAccept[系统向数据库提交接受响应] Choose -- 拒绝交易 --> SystemReject[系统向数据库提交拒绝响应] SystemAccept --> Decision{ } SystemReject --> Decision Decision -- 结束 --> End(()) Decision -- 是否继续? --> Continue{ } Continue -- 是 --> Enter Continue -- 否 --> End </pre>

用例名称： 查看交易请求
参与者： 学生或管理员
前置条件： 学生或管理员进入交易中介模块界面，查看交易请求界面
后置操作： 展示参与者可见的交易请求信息
工作流程：

<ol style="list-style-type: none"> 1. 学生或管理员进入交易请求界面 2. 系统根据账号信息向数据库查询交易请求的信息 3. 若为学生账号，则返回该学生所有的交易请求 4. 若为管理员账号，则返回所有的交易请求与相关信息 5. 系统展示交易请求信息列表
扩展流程： 如果步骤1时学生或管理员未登录，则进入登录界面，待用户登录后继续后续步骤

用例名称： 筛选交易请求
参与者： 学生或管理员
前置条件： 学生或管理员进入交易中介模块界面，查看交易请求界面
后置操作： 展示筛选后的交易请求信息
工作流程： <ol style="list-style-type: none"> 1. 学生或管理员进入交易请求界面 2. 系统根据账号信息向数据库查询交易请求的信息 3. 系统返回交易请求的信息列表 4. 学生或管理员选择对信息列表进行筛选 5. 学生或管理员提交筛选选项 6. 系统接收筛选选项，对数据库查询返回信息进行筛选 7. 系统展示筛选后的交易请求信息列表
扩展流程： 如果步骤1时学生或管理员未登录，则进入登录界面，待用户登录后继续后续步骤

用例名称： 删除交易请求
参与者： 学生或管理员
前置条件： 学生或管理员进入交易中介模块界面，查看交易请求界面
后置操作： 交易请求被删除并更新状态
工作流程： <ol style="list-style-type: none"> 1. 学生或管理员进入交易请求界面 2. 系统根据账号信息向数据库查询交易请求的信息 3. 系统返回交易请求的信息列表 4. 学生或管理员选择要删除的交易请求 5. 系统提示确认删除操作 6. 学生或管理员确认删除操作 7. 系统接收删除请求，向数据库发送删除操作请求 8. 若删除成功，系统返回删除成功提示，更新交易请求列表 9. 若删除失败，系统返回失败提示，返回步骤2
扩展流程： 如果步骤1时学生或管理员未登录，则进入登录界面，待用户登录后继续后续步骤

用例名称： 修改交易信息
参与者： 学生或管理员
前置条件： 学生或管理员进入交易中介模块界面，查看交易请求界面
后置操作： 更新交易信息

工作流程： <ol style="list-style-type: none"> 1. 学生或管理员进入交易请求界面 2. 系统根据账号信息向数据库查询交易请求的信息 3. 系统返回交易请求的信息列表 4. 学生或管理员选择要修改的交易信息 5. 学生或管理员输入修改后的交易信息 6. 学生或管理员提交修改 7. 系统接受修改，向数据库发送修改请求 8. 若修改成功，系统返回修改成功提示，更新交易信息 9. 若修改失败，系统返回失败提示，返回步骤2
扩展流程： 如果步骤1时学生或管理员未登录，则进入登录界面，待用户登录后继续后续步骤

用例名称： 管理交易请求
参与者： 学生或管理员
前置条件： 学生或管理员进入交易中介模块界面，查看交易请求界面
后置操作： 根据学生或管理员进行的操作更新交易信息
工作流程： <ol style="list-style-type: none"> 1. 学生或管理员进入交易请求界面 2. 系统根据账号信息向数据库查询交易请求的信息 3. 系统返回交易请求的信息列表 4. 学生或管理员选择要进行的操作(如删除，修改等) 5. 系统接收操作请求执行对应操作 6. 系统根据操作结果返回成功或失败提示
扩展流程： 如果步骤1时学生或管理员未登录，则进入登录界面，待用户登录后继续后续步骤
活动图： <pre> graph LR Start(()) --> EnterView[进入查看交易请求界面] EnterView --> IsAdmin{是否为管理员?} IsAdmin -- 是 --> ReturnAll[系统返回全部交易请求列表] IsAdmin -- 否 --> ReturnVisible[系统返回该学生可见的交易请求列表] ReturnAll --> SelectRequest[选择一个待管理的交易请求] ReturnVisible --> SelectRequest SelectRequest --> SelectOp{选择管理操作?} SelectOp -- 选择“修改” --> SubmitMod[系统提交修改请求和信息] SelectOp -- 选择“删除” --> SubmitDel[系统提交删除请求和信息] SelectOp -- 选择“确定” --> RequestList[系统请求并返回待选交易列表] SubmitMod --> IsEnded{是否结束?} SubmitDel --> IsEnded RequestList --> IsEnded IsEnded -- 是 --> End((结束)) IsEnded -- 否 --> ReturnAll IsEnded -- 否 --> ReturnVisible End --> Exit[退出] Exit --> End2(()) </pre>

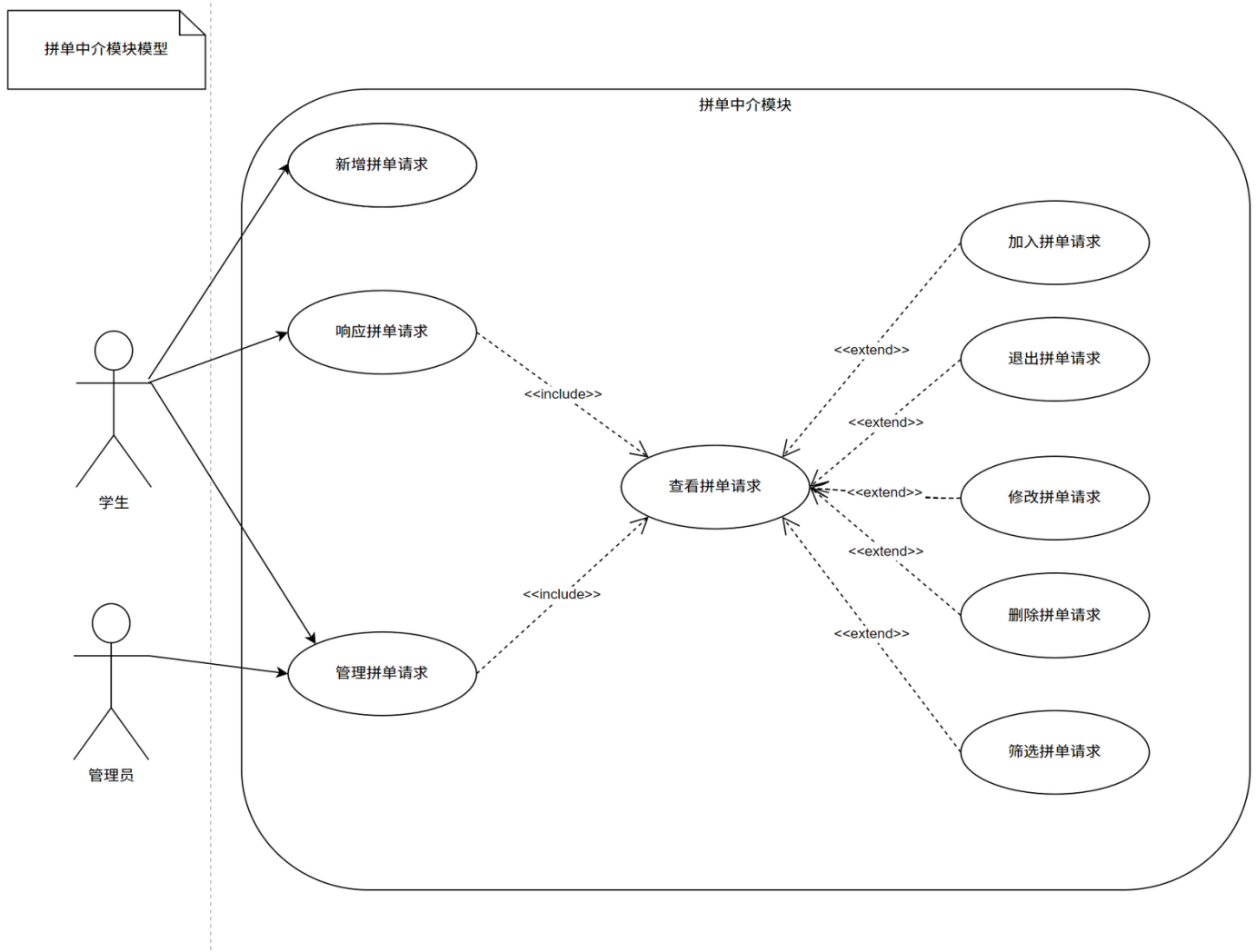
2.3 拼单中介模块

描述： <p>模块功能：模块处理拼单请求的管理与交互，包括新增、响应、管理、查看等功能。</p> <p>输入：学生或管理员输入的操作请求，拼单请求的具体信息。</p>
--

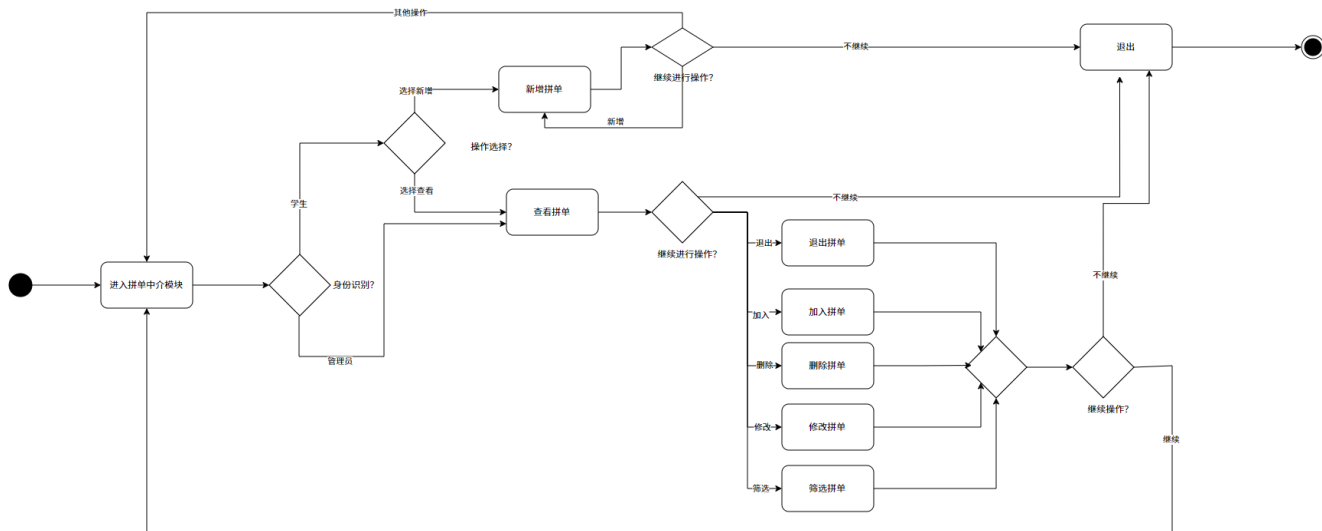
输出：拼单请求状态更新情况信息与确认信息。

模块边界：模块与用户登录部分和数据库交互，负责学生与管理员的账号数据以及拼单请求数据的存取与修改。

模块用例图：



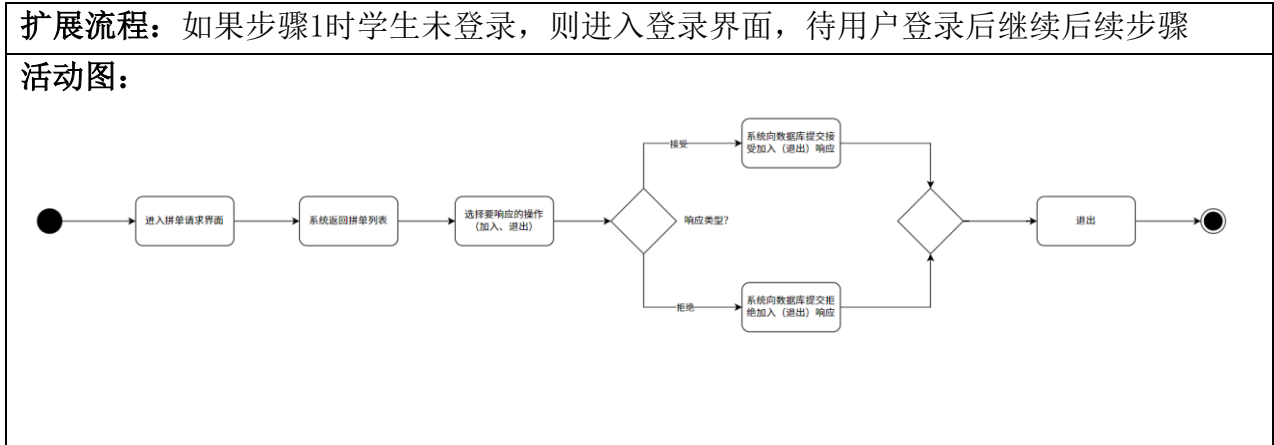
模块活动图：



用例规约：

用例名称： 新增拼单请求
参与者： 学生
前置条件： 学生进入拼单中介模块界面
后置操作： 更新新增拼单信息
工作流程： <ol style="list-style-type: none"> 1. 学生选择“新增拼单请求”选项 2. 学生输入拼单请求信息 3. 学生提交拼单请求 4. 系统验证输入信息的完整性 5. 若消息完整，系统保存拼单请求，返回新增成功消息 6. 若消息不完整，系统提示错误，返回步骤2
扩展流程： 如果步骤1时学生未登录，则进入登录界面，待用户登录后继续后续步骤

用例名称： 响应拼单请求
参与者： 学生
前置条件： 学生进入拼单中介模块界面，查看拼单请求界面
后置操作： 响应拼单请求状态更新
工作流程： <ol style="list-style-type: none"> 1. 学生进入拼单请求界面 2. 系统根据学生账号信息向数据库查询拼单请求的信息 3. 系统返回拼单请求的信息列表 4. 学生选择一个需要的拼单请求进行响应 5. 学生选择对应的响应操作(加入拼车单，退出拼车单) 6. 学生提交响应信息 7. 系统接受响应信息 8. 若响应成功，系统返回响应成功消息，并向数据库更新信息 9. 若响应不成功，系统提示错误信息，返回步骤2



用例名称： 查看拼单请求
参与者： 学生、管理员
前置条件： 学生或管理员进入拼单中介模块界面，查看交易请求界面
后置操作： 展示参与者可见的拼单请求信息
工作流程： <ol style="list-style-type: none"> 1. 学生或管理员进入拼单请求界面 2. 系统根据账号信息向数据库查询交易请求的信息 3. 若为学生账号，则返回该学生所有的拼单请求 4. 若为管理员账号，则返回所有的拼单请求与相关信息 5. 系统展示拼单请求信息列表
扩展流程： 如果步骤1时学生或管理员未登录，则进入登录界面，待用户登录后继续后续步骤

用例名称： 筛选拼单请求
参与者： 学生、管理员
前置条件： 学生或管理员进入拼单中介模块界面，查看拼单请求界面
后置操作： 展示筛选后的拼单请求信息
工作流程： <ol style="list-style-type: none"> 1. 学生或管理员进入拼单请求界面 2. 系统根据账号信息向数据库查询拼单请求的信息 3. 系统返回拼单请求的信息列表 4. 学生或管理员选择对信息列表进行筛选 5. 学生或管理员提交筛选选项 6. 系统接收筛选选项，对数据库查询返回信息进行筛选 7. 系统展示筛选后的拼单请求信息列表
扩展流程： 如果步骤1时学生或管理员未登录，则进入登录界面，待用户登录后继续后续步骤

用例名称： 删除拼单请求
参与者： 学生或管理员
前置条件： 学生或管理员进入拼单中介模块界面，查看拼单请求界面
后置操作： 拼单请求被删除并更新状态
工作流程：

<ol style="list-style-type: none"> 1. 学生或管理员进入拼单请求界面 2. 系统根据账号信息向数据库查询拼单请求的信息 3. 系统返回拼单请求的信息列表 4. 学生或管理员选择要删除的拼单请求 5. 系统提示确认删除操作 6. 学生或管理员确认删除操作 7. 系统接收删除请求，向数据库发送删除操作请求 8. 若删除成功，系统返回删除成功提示，更新拼单请求列表 9. 若删除失败，系统返回失败提示，返回步骤2
扩展流程： 如果步骤1时学生或管理员未登录，则进入登录界面，待用户登录后继续后续步骤

用例名称： 修改拼单请求
参与者： 学生或管理员
前置条件： 学生或管理员进入拼单中介模块界面，查看拼单请求界面
后置操作： 更新拼单信息
工作流程： <ol style="list-style-type: none"> 1. 学生或管理员进入拼单请求界面 2. 系统根据账号信息向数据库查询拼单请求的信息 3. 系统返回拼单请求的信息列表 4. 学生或管理员选择要修改的拼单信息 5. 学生或管理员输入修改后的拼单信息 6. 学生或管理员提交修改 7. 系统接受修改，向数据库发送修改请求 8. 若修改成功，系统返回修改成功提示，更新拼单信息 9. 若修改失败，系统返回失败提示，返回步骤2
扩展流程： 如果步骤1时学生或管理员未登录，则进入登录界面，待用户登录后继续后续步骤

用例名称： 加入拼单请求
参与者： 学生
前置条件： 学生进入拼单中介模块界面，查看拼单请求界面
后置操作： 更新拼单信息
工作流程： <ol style="list-style-type: none"> 1. 学生进入拼单请求界面 2. 系统根据学生账号信息向数据库查询拼单请求的信息 3. 系统返回拼单请求的信息列表 4. 学生选择一个需要加入的拼单 5. 学生提交需要加入的拼单信息 6. 系统接受加入请求 7. 若加入成功，系统返回加入成功消息，并向数据库更新信息 8. 若加入不成功，系统提示错误信息，返回步骤2
扩展流程： 如果步骤1时学生未登录，则进入登录界面，待用户登录后继续后续步骤

用例名称： 退出拼单请求
参与者： 学生
前置条件： 学生进入拼单中介模块界面，查看拼单请求界面
后置操作： 更新拼单信息
工作流程： 1. 学生进入拼单请求界面 2. 系统根据学生账号信息向数据库查询拼单请求的信息 3. 系统返回拼单请求的信息列表 4. 学生选择一个需要退出的拼单 5. 学生提交需要退出的拼单信息 6. 系统接受退出请求 7. 若退出成功，系统返回加入退出消息，并向数据库更新信息 8. 若退出不成功，系统提示错误信息，返回步骤2
扩展流程： 如果步骤1时学生未登录，则进入登录界面，待用户登录后继续后续步骤

用例名称： 管理拼单请求
参与者： 学生或管理员
前置条件： 学生或管理员进入拼单中介模块界面，查看拼单请求界面
后置操作： 根据学生或管理员进行的操作更新拼单信息
工作流程： 1. 学生或管理员进入拼单请求界面 2. 系统根据账号信息向数据库查询拼单请求的信息 3. 系统返回拼单请求的信息列表 4. 学生或管理员选择要进行的操作(如删除，修改，加入，退出等) 5. 系统接收操作请求执行对应操作 6. 系统根据操作结果返回成功或失败提示
扩展流程： 如果步骤1时学生或管理员未登录，则进入登录界面，待用户登录后继续后续步骤

2.4 信息查阅模块

描述：

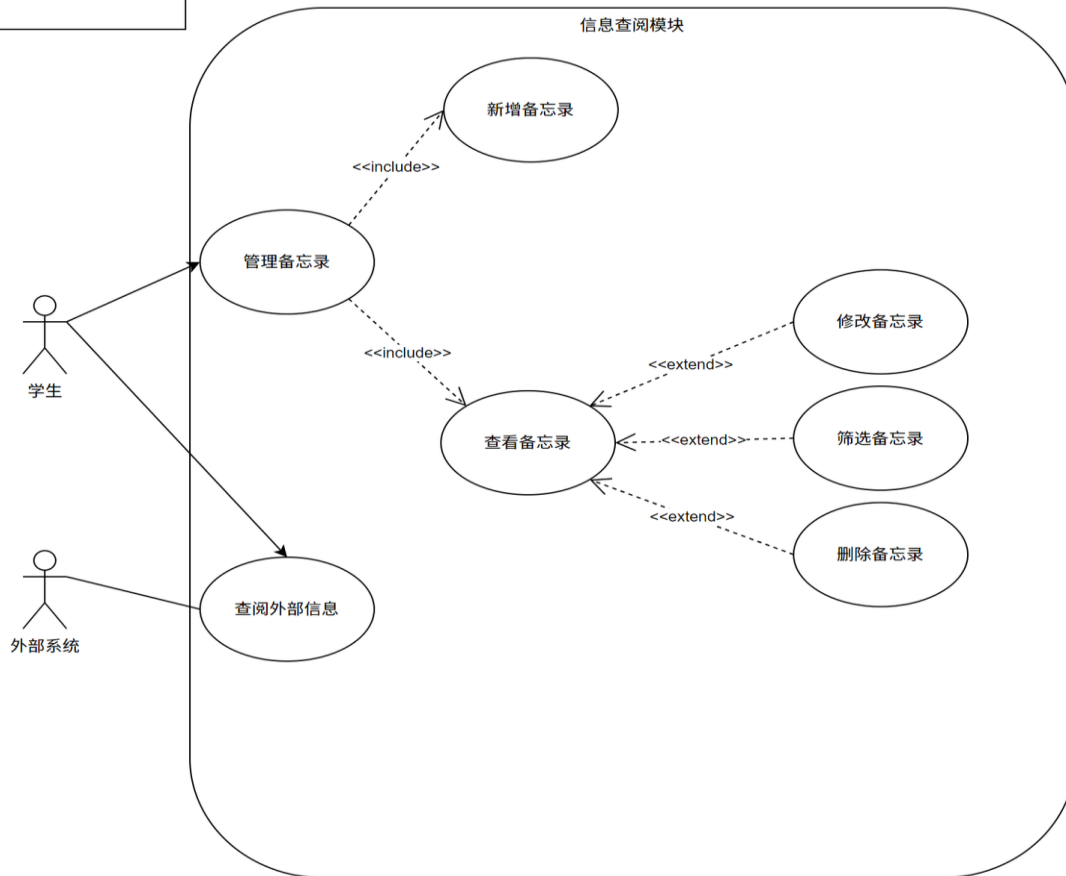
模块功能：管理个人备忘录，连接和查阅外部系统信息。

输入：学生对于备忘录修改的具体信息；选择外部系统。

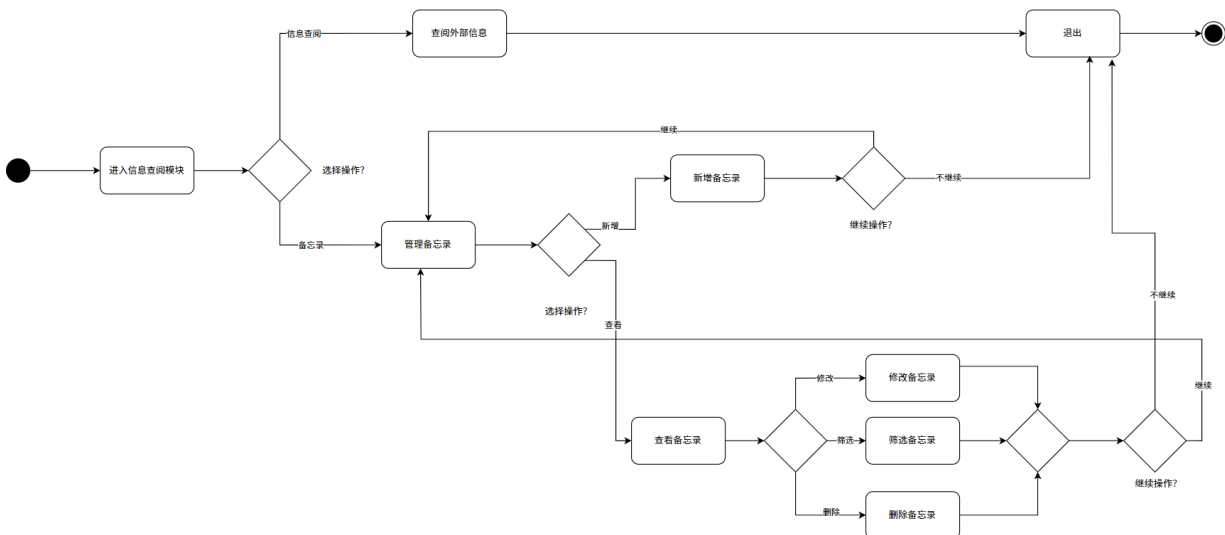
输出：备忘录信息更新；跳转外部系统。

模块边界：模块与用户登录部分和数据库以及外部系统交互，负责学生备忘录的管理操作，提供外部系统接口。

模块用例图：



模块活动图：



用例规约：

用例名称：查阅外部信息
参与者：学生、外部系统
前置条件：学生进入信息查阅模块
后置操作：跳转至外部系统

工作流程： 1. 学生选择所需查阅的外部系统 2. 学生提交查阅请求 3. 若成功，系统保存当前状态，跳转至外部系统 4. 若失败，系统提示错误，返回步骤1
扩展流程： 无

用例名称： 管理备忘录
参与者： 学生
前置条件： 学生进入信息查阅模块
后置操作： 根据操作更新备忘录信息
工作流程： 1. 学生进入信息查阅界面 2. 系统根据账号信息向数据库查询学生备忘录的信息 3. 系统返回学生备忘录列表 4. 学生选择要进行的操作(如新增，修改，筛选，删除等) 5. 系统接收操作请求执行对应操作 6. 系统根据操作结果返回成功或失败提示
扩展流程： 如果步骤1时学生未登录，则进入登录界面，待用户登录后继续后续步骤
活动图： <pre> graph LR Start(()) --> Enter[进入信息查询界面] Enter --> SelectOp1{选择操作?} SelectOp1 -- 新增 --> AddMemo[新增备忘录] AddMemo --> Exit[退出] SelectOp1 -- 查看 --> ReturnList[系统返回备忘录列表] ReturnList --> SelectOp2{选择操作?} SelectOp2 -- 修改 --> SubmitMod[系统提交修改请求和信息] SelectOp2 -- 删除 --> SubmitDel[系统提交删除请求和信息] SelectOp2 -- 筛选 --> SubmitFilter[系统提交请求并返回筛选列表] SubmitMod --> Merge{ } SubmitDel --> Merge SubmitFilter --> Merge Merge --> Exit Exit --> End(()) </pre>

用例名称： 新增备忘录
参与者： 学生
前置条件： 学生进入信息查阅模块
后置操作： 更新备忘录
工作流程： 1. 学生选择“新增备忘录”选项 2. 学生输入备忘录信息 3. 学生提交备忘录信息 4. 系统保存备忘录信息，返回新增成功
扩展流程： 如果步骤1时学生未登录，则进入登录界面，待用户登录后继续后续步骤

用例名称： 查看备忘录
参与者： 学生
前置条件： 学生进入信息查阅模块

后置操作： 展示该学生所有备忘录
工作流程： <ol style="list-style-type: none"> 1. 学生进入信息查询界面 2. 系统根据账号信息向数据库查询备忘录信息 3. 系统返回该学生所有的备忘录信息 4. 系统展示备忘录信息
扩展流程： 如果步骤1时学生未登录，则进入登录界面，待用户登录后继续后续步骤

用例名称： 修改备忘录
参与者： 学生
前置条件： 学生进入信息查阅模块，查看备忘录界面
后置操作： 更新备忘录
工作流程： <ol style="list-style-type: none"> 1. 学生进入信息查询界面 2. 系统根据账号信息向数据库查询备忘录信息 3. 系统返回所有备忘录信息 4. 学生选择要修改的备忘录 5. 学生输入修改内容 6. 学生提交修改内容 7. 系统接受修改，向数据库发送修改请求 8. 若修改成功，系统返回修改成功提示，更新备忘录信息 9. 若修改失败，系统返回失败提示，返回步骤2
扩展流程： 如果步骤1时学生未登录，则进入登录界面，待用户登录后继续后续步骤

用例名称： 筛选备忘录
参与者： 学生
前置条件： 学生进入信息查阅模块，查看备忘录界面
后置操作： 展示筛选后的备忘录信息
工作流程： <ol style="list-style-type: none"> 1. 学生进入信息查询界面 2. 系统根据账号信息向数据库查询备忘录信息 3. 系统返回所有备忘录信息 4. 学生选择对信息列表进行筛选 5. 学生提交筛选选项 6. 系统接收筛选选项，对数据库查询返回信息进行筛选 7. 系统展示筛选后的交易请求信息列表
扩展流程： 如果步骤1时学生未登录，则进入登录界面，待用户登录后继续后续步骤

用例名称： 删除备忘录
参与者： 学生
前置条件： 学生进入信息查阅模块，查看备忘录界面
后置操作： 指定备忘录被删除并更新备忘录信息表

工作流程： <ol style="list-style-type: none"> 1. 学生进入信息查询界面 2. 系统根据账号信息向数据库查询备忘录信息 3. 系统返回所有备忘录信息 4. 学生选择要删除的备忘录信息 5. 系统提示确认删除操作 6. 学生确认删除操作 7. 系统接收删除请求，向数据库发送删除操作请求 8. 若删除成功，系统返回删除成功提示，更新备忘录信息表 9. 若删除失败，系统返回失败提示，返回步骤2
扩展流程： 如果步骤1时学生未登录，则进入登录界面，待用户登录后继续后续步骤

2.5 敏捷需求表

考虑到在项目计划中提到项目的设计与开发将涉及到敏捷模型，此处我们预计使用Scrum敏捷模型，从已有的上述需求出发，在开发的过程中逐步完善需求并燃尽需求。以下是目前已知需求的敏捷需求表。

用户故事	工作量 (人·日)	优先级 (1-10)	预计完成的 Sprint
学生希望能够在我们的系统中达成与其他同学进行交易的功能，涉及到创建、修改、响应请求几个子功能	40	10	第一
学生希望能够在我们的系统中达成与其他同学一起拼单的功能，涉及到创建、修改、响应请求几个子功能	45	8	第二
学生希望能够在我们的系统中导航到外部系统或引入外部系统的信息来查看自己相关的信息	10	4	未定
学生希望能够在我们的系统中记录一个自己的备忘录以供查阅	15	4	未定
管理员希望能够在我们的系统中管理学生提出的交易或者拼单请求，能够进行关闭和监督等操作	25	6	未定

3. 术语表

拼单： 指多个用户共同购买某一商品以享受折扣或降低运费的交易方式，用户可以发布拼单信息并根据条件筛选匹配其他参与者。
外部系统： 指与本平台进行数据交互或功能集成的其他系统，如学校的选课系统、成绩查询系统等，提供用户所需的额外信息和服务。
绩点： 学生在课程学习中所获得的学术成绩的量化指标，通常用于评估学生的学业表现和排名。
备忘录： 学生在课程学习中所获得的学术成绩的量化指标，通常用于评估学生的学业表现和排名。
课程： 学校为学生提供的学习内容单元，通常包含讲授知识、课外作业和考试等，是学生学业的一部分。
交易匹配： 指系统根据用户发布的商品信息和需求自动匹配合适的交易方，确保交易的顺利进行和双方的需求得到满足。
用户注册： 用户在平台上创建账户的过程，通过提供基本信息（如邮箱、密码等）以获得访问平台功能的权限。
信息查询： 用户通过平台获取课程信息、交易记录、拼单信息等服务的过程，旨在提高用户获取信息的效率。
系统集成： 将多个独立的系统或模块连接在一起，以实现数据共享和功能互通，提高整体服务的连贯性和用户体验。
消息通知： 平台向用户推送的实时信息或提醒，用于通知用户有关交易、拼单、课程更新等重要信息，增强用户互动。
校园交流： 指同学之间通过平台进行信息共享和互动的过程，促进同学间的沟通与合作。
交易管理： 指在平台上进行商品发布、交易记录跟踪、支付处理和纠纷解决的综合管理功能，确保交易过程的顺利和安全。
用户体验（UX）： 指用户在使用平台过程中的整体感受，包括界面设计、操作流畅度和功能可用性等，直接影响用户对平台的满意度。
系统接口： 指平台与外部系统或服务之间的数据交互和功能调用的通道，包括API接口、用户界面和硬件接口。
课程评价： 学生对所选课程进行反馈的机制，通常包括对课程内容、教师教学和学习体验的评分与评论。
交易安全： 指在平台上进行交易时，保护用户信息和资金安全的机制，包括加密传输、身份验证和交易监控等。
数据分析： 通过收集和分析用户数据，识别用户行为和需求的过程，以优化平台功能和提升用户体验。

4. 补充规约

4.1 用户类型

1, 学生

- **权限与特征:** 学生为经过学生身份信息认证的用户。可以使用交易拼单、信息查询、课程选课等基本功能。用户信息和交易记录仅限于个人可见, 保护隐私。
- **使用模块:** 交易模块、拼单模块、信息查询模块、备忘录功能等。
- **性能体验关注点:** 关注系统的响应时间和并发处理能力, 确保在高流量情况下也能顺畅使用, 尤其是在交易高峰期。

2, 管理员

- **权限与特征:** 管理员拥有更高的权限, 可以管理用户信息、审核交易信息、处理用户反馈等。可以访问系统后台, 查看系统运行状态和用户数据。
- **使用模块:** 后台管理模块、用户管理模块、数据统计模块。
- **性能体验关注点:** 关注系统的稳定性和数据处理能力, 以确保管理功能的顺利进行。

4.2 环境描述

本系统将部署在B/S（浏览器/服务器）架构下, 用户通过现代浏览器访问平台, 支持主流操作系统如Windows和Linux端的浏览器访问。前端部分将使用Vue.js构建, 后端使用ASP.NET和C#开发, 整个系统将在云服务器上运行, 以确保高可用性和可扩展性。数据存储将采用Oracle数据库, 系统将利用Docker容器化技术来简化部署和管理, 保证不同环境间的一致性。同时, Nginx将作为反向代理服务器, 负责处理客户端请求和负载均衡, 以提升系统性能和安全性。整体环境配置将确保系统在高并发情况下依然保持稳定与高效。

4.3 假设、限制与依赖

假设:

系统将部署在学校服务器或云服务器上, 确保高可用性和可靠性。

所有第三方API和服务的授权和接口调用已获得相关方的许可。

平台用户量将在一个可控的范围内逐步增长, 初期用户增长将不会对系统造成负载过大。

限制:

系统需支持至少1000名学生的并发访问, 确保在高峰期间性能不降低。

系统运行前需要执行定时任务, 如每日数据备份和安全扫描, 以确保系统正常运行。

系统的响应时间应控制在3秒以内, 以确保良好的用户体验。

系统需能够兼容多种设备（如PC、手机、平板）和浏览器（如Chrome、Firefox、Edge）。

由于数据安全和隐私保护的要求, 所有用户数据需进行加密存储和传输。

依赖:

外部系统依赖: 系统需要与学校的教学管理信息系统、选课系统等进行集成。

技术依赖: 系统依赖于Vue框架进行前端开发, 使用Axios进行HTTP请求, 后端使用ASP.NET和C#进行逻辑处理。

接口依赖: 系统依赖于RESTful API与外部系统进行数据交换。

开发工具依赖: 项目开发中使用的工具和技术栈如GitHub、Docker、Oracle数据库、Nginx服务

器等。

第三方服务依赖：某些功能可能需要第三方服务支持（如支付网关、短信验证等）。

持续集成和部署（CI/CD）依赖：依赖于GitHub Actions等CI/CD工具，实现自动化测试和部署，提高开发效率和可靠性。

4.4 系统接口

界面接口：用户通过浏览器访问系统的各项服务，支持的浏览器包括Chrome、Firefox、Edge等现代浏览器。

硬件接口：系统前端使用Vue框架，后端使用ASP.NET框架，通过HTTP协议与服务器通信。使用Nginx作为反向代理服务器，开放80和443端口处理HTTP和HTTPS请求。

外部接口：系统与学校的教学管理信息系统、选课系统等外部系统通过同济大学开放平台API进行数据交换。由于使用接口较多，仅举例如下：

查询学生所有考试信息接口：

描述：获取ltongji系统上学生所有的考试信息

支持的模式：client_credentials authorization_code

请求方法：GET

请求地址：https://api.tongji.edu.cn/v1/rt/onetongji/student_exams

数据来源：教务系统

请求参数：

参数名：calendarId 类型：integer 备注：学期编号，为空默认为当前学期编号；通过“查询所有学期日历编号”获取历史学期编号

参数名：examType 类型：integer 备注：考试类型，1为正常考试，2为缓考。为空默认为正常考试

参数名：userId 类型：string 备注：学生学号

查询本科生课程成绩信息接口：

描述：获取ltongji系统上本科生课程的成绩信息

支持的模式：client_credentials authorization_code

请求方法：GET

请求地址：https://api.tongji.edu.cn/v1/rt/onetongji/student_exams

数据来源：教务系统

请求参数：

参数名：calendarId 类型：integer 备注：学期编号，为空默认为当前学期编号；通过“查询所有学期日历编号”获取历史学期编号；-1返回所有学期的成绩

参数名：userId 类型：string 备注：学生学号

4.5 非功能性需求

性能要求：系统需支持至少同时1000名学生的并发访问，响应时间应控制在3秒以内。数据库查询应在1000毫秒内完成，以确保系统的响应速度合适。

安全要求：系统需采用HTTPS传输，以保护用户数据在传输过程中的安全。数据库需定期备份，用户数据需进行加密存储。系统需具备防止SQL注入、跨站脚本攻击（XSS）等安全措施。用户登录需支持验证码等多次身份验证，以提高账户的安全性。

可维护性：系统代码需具备良好的可读性，遵循编码规范，采用模块化设计，方便后期维护和扩展。提供详细的开发文档和用户手册，帮助开发者和用户快速上手。

扩展性：系统设计需具备高扩展性，支持功能模块的快速增加和更新。前后端分离架构，允许前端和后端独立开发和部署，以提高开发效率和系统扩展性。

可靠性：系统需具备高可用性，保证99%的正常运行时间（Uptime）。提供实时监控和报警机制，及时发现和处理系统异常。

可用性：系统界面需简洁直观，提供一致的用户体验，提供详细的错误提示信息，帮助用户理解和解决操作中的问题。

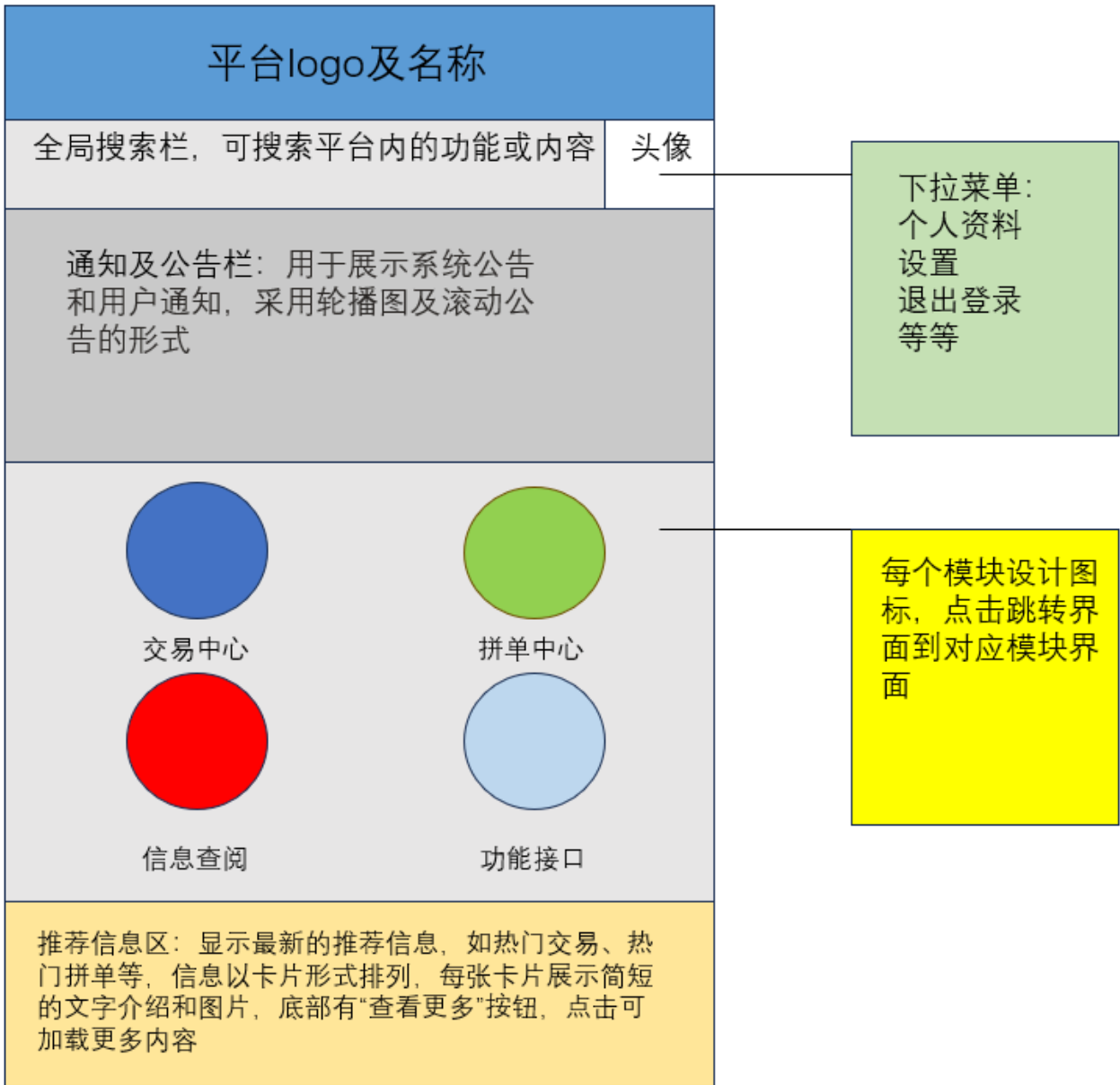
可移植性：采用容器化技术（如Docker），方便系统的迁移和部署。

可恢复性：系统需提供快速的备份和恢复方案，以应对数据丢失或系统故障。

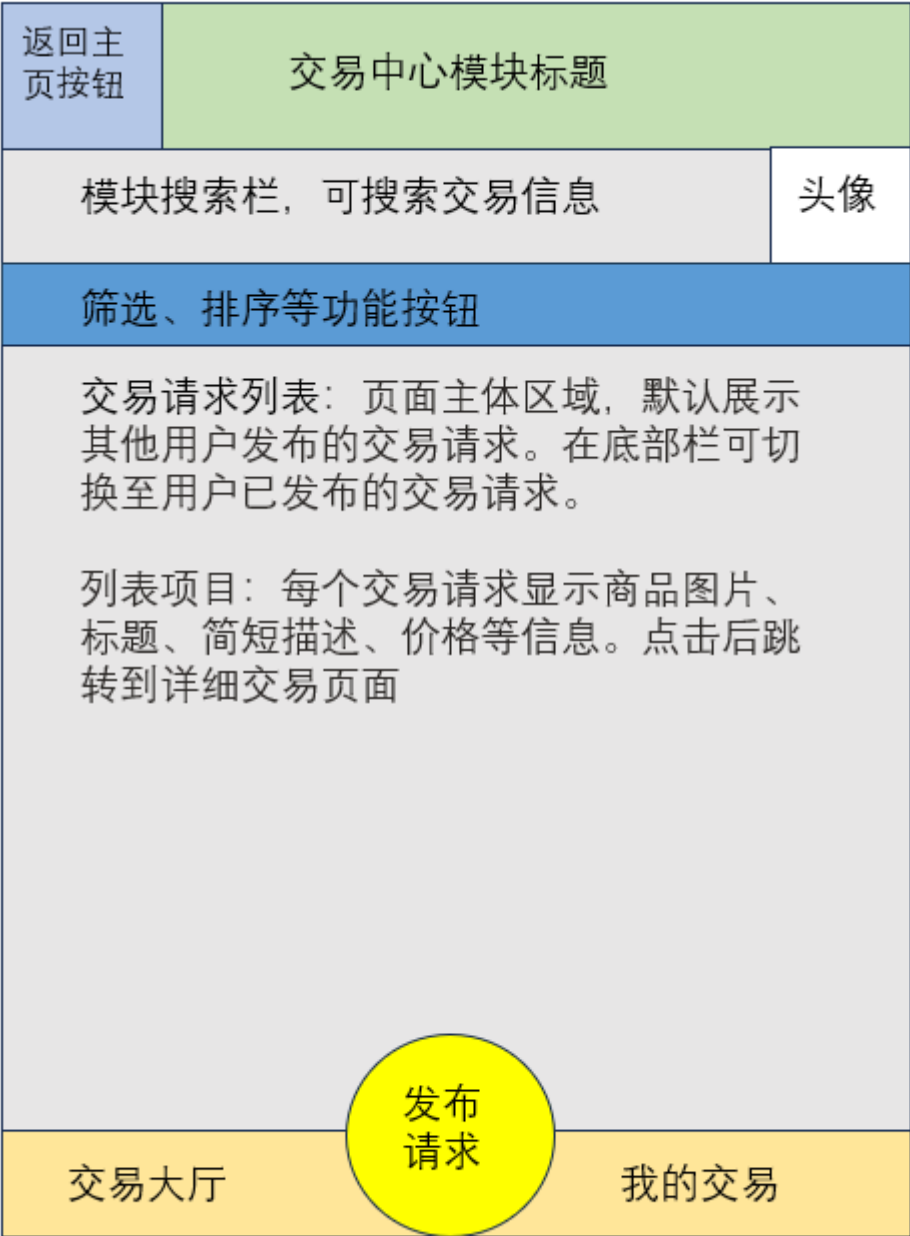
用户体验（UX）：系统界面设计需符合用户习惯，操作简单方便，减少学习成本。提供个性化定制功能，根据用户需求提供个性化服务。

5. 初始快照

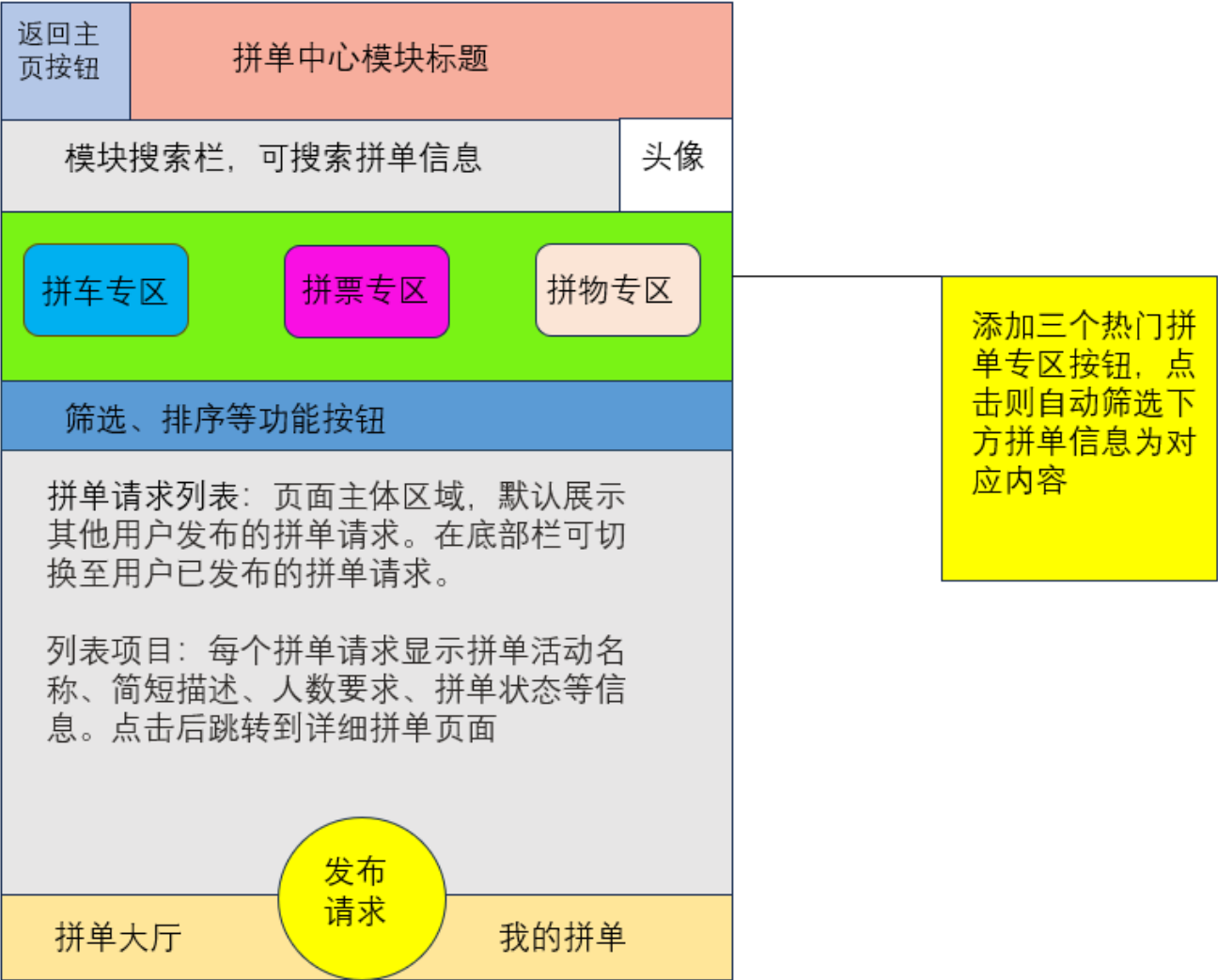
5.1 主页快照



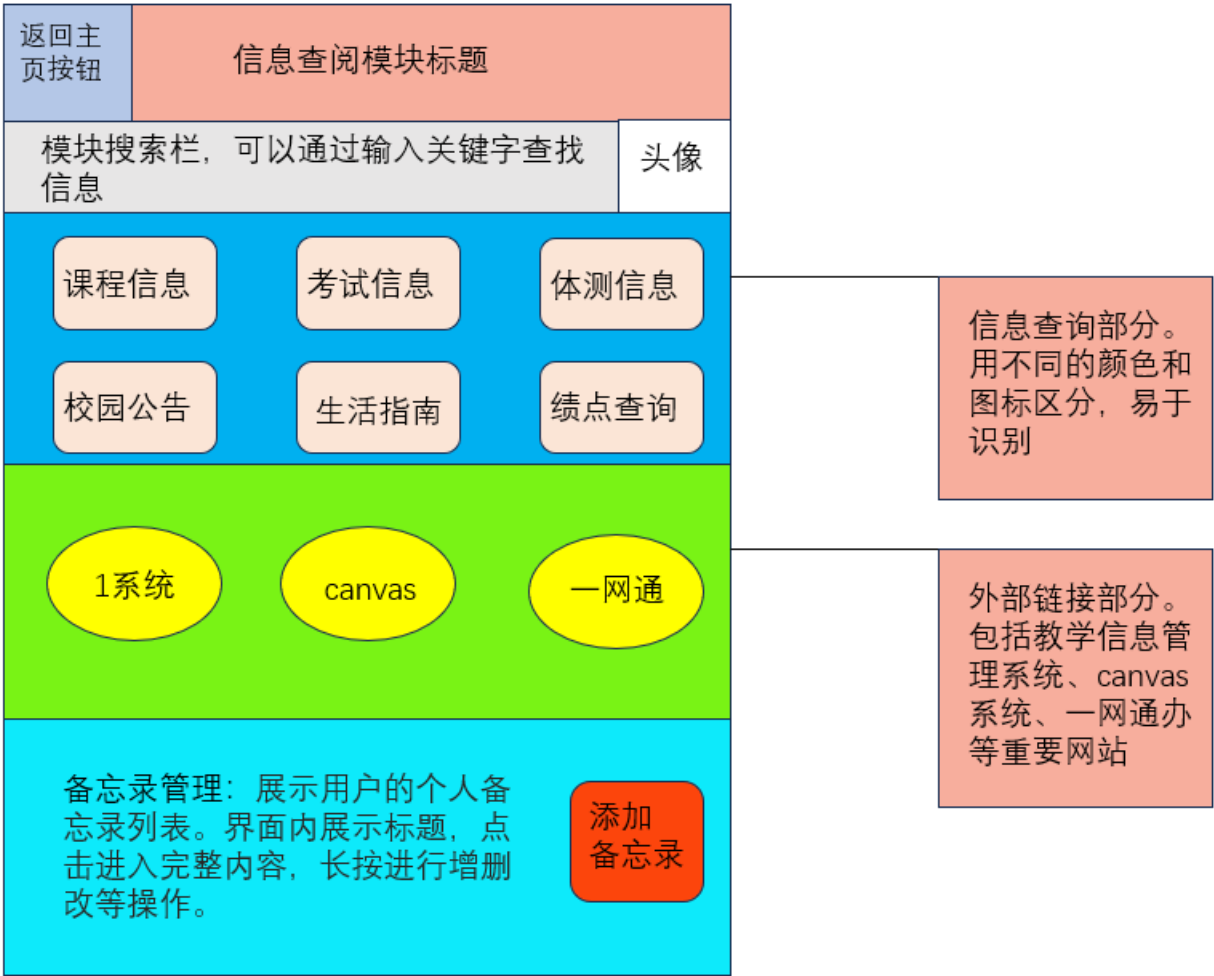
5.2 交易中介模块快照



5.3 拼单中介模块快照



5.4 信息查阅模块快照



贡献列表

王天阔：30%，包括功能性需求、顶层用例和活动的建模，以及补充规约。

刘骏伟：30%，包括交易中介模块用例和活动建模，以及补充规约。

倪佳瑞：40%，包括拼单中介模块、信息查阅模块的用例和活动建模，以及补充规约。

参考资料

相关的软件工程资料，需求分析方法资料、各类课件或指导教材、可借鉴的相似项目的资料、同济开放平台的API规范等等文件。

[1] 《软件工程：实践者的研究方法 第9版》，ISBN-978-7-111-68394-0。该书也是软件工程课程的课程教材之一，讲述了作为软件开发的设计者、管理者和开发者应当学习和遵循的一套规范方法。在本书中讲述了敏捷模型和软件开发的过程模式等内容。我们借鉴其中的一部分内容，作为我们在分析需求和设计开发过程的指导。

[2] <http://api.tongji.edu.cn/docs/#/>，同济大学开放平台说明文档。该平台是由同济大学官方提供的供非学校官方系统获取官方数据库中信息的可靠接口，被已被多种学生开发的平台所使用。由于我们的系统中有接入外部系统的需求，故参考该API规范是有必要的。

[3] 王继成, 高珍. 软件需求分析的研究[J]. 计算机工程与设计, 2002, (08):18-21. DOI:10.16208/j.issn1000-7024.2002.08.005. 该期刊论文是由同济大学王继成和高珍撰写的，主要内容即为软件工程需求分析中的相关方法 and 研究，对于指导我们形成本文档有重要的作用。