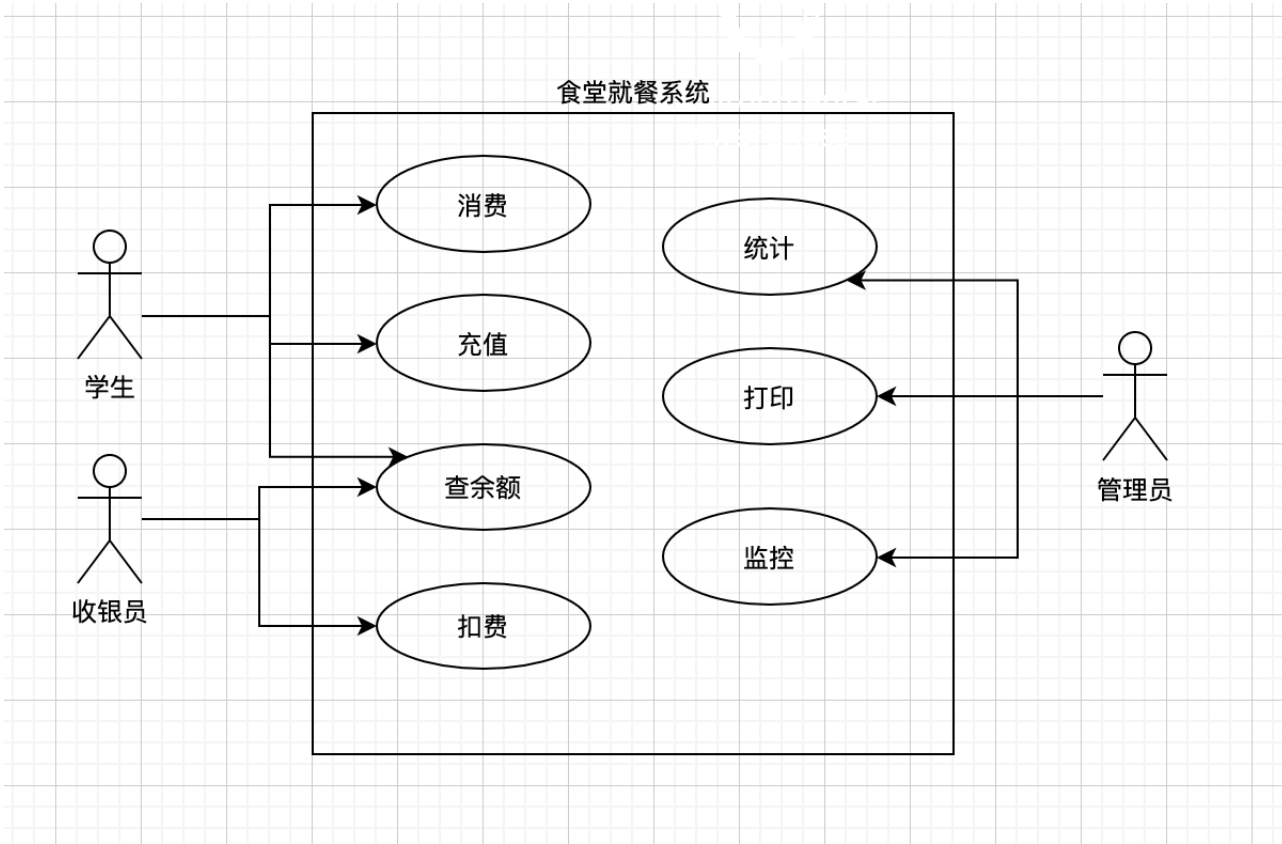


食堂就餐卡系统设计

1. 设计概述

在学校中，为了能够使学生就餐时可以方便结算，设计了如下图所示的系统。



1.1 功能概述

这个系统的主要使用者包括学生，食堂收银员以及系统的管理员。

学生可以注册充值，查看余额，消费等，食堂的收银员可以通过系统查看卡的余额，输入金额并进行扣款，而在后台，管理员可以对整个系统进行监控，看到并打印每一笔消费的流水等。

1.2 非功能约束

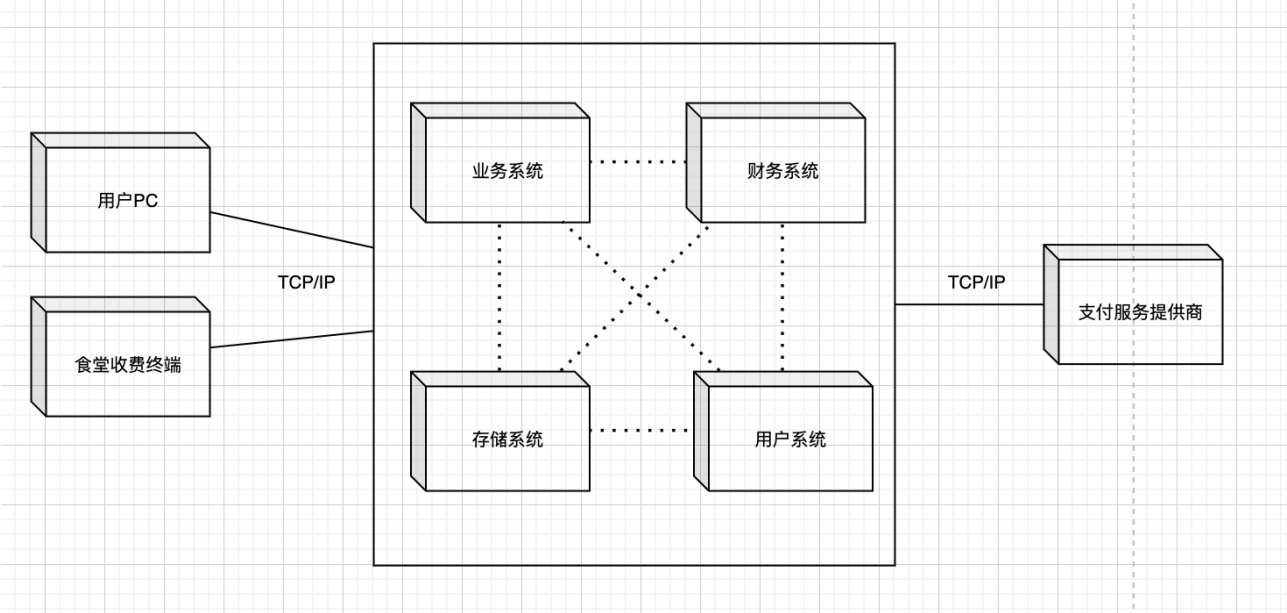
系统的用户量大约为1万人，每日预计产生约3万笔消费，10万次左右的查询。

1. 查询性能目标：平均响应时间 < 600ms，95%响应时间 < 2000ms，单机TPS > 100
2. 消费性能目标：平均响应时间 < 1000ms，95%响应时间 < 2000ms，单机TPS > 30
3. 充值性能目标：平均响应时间 < 2000ms，95%响应时间 < 3000ms，单机TPS > 30
4. 系统核心功能可用性目标：> 99.9%
5. 数据持久化目标：> 99.99999%

2. 系统部署图与整体设计

系统上线时预计部署9台物理机，3个子系统，和一个外部支付供应商进行交互。

2.1 系统部署图



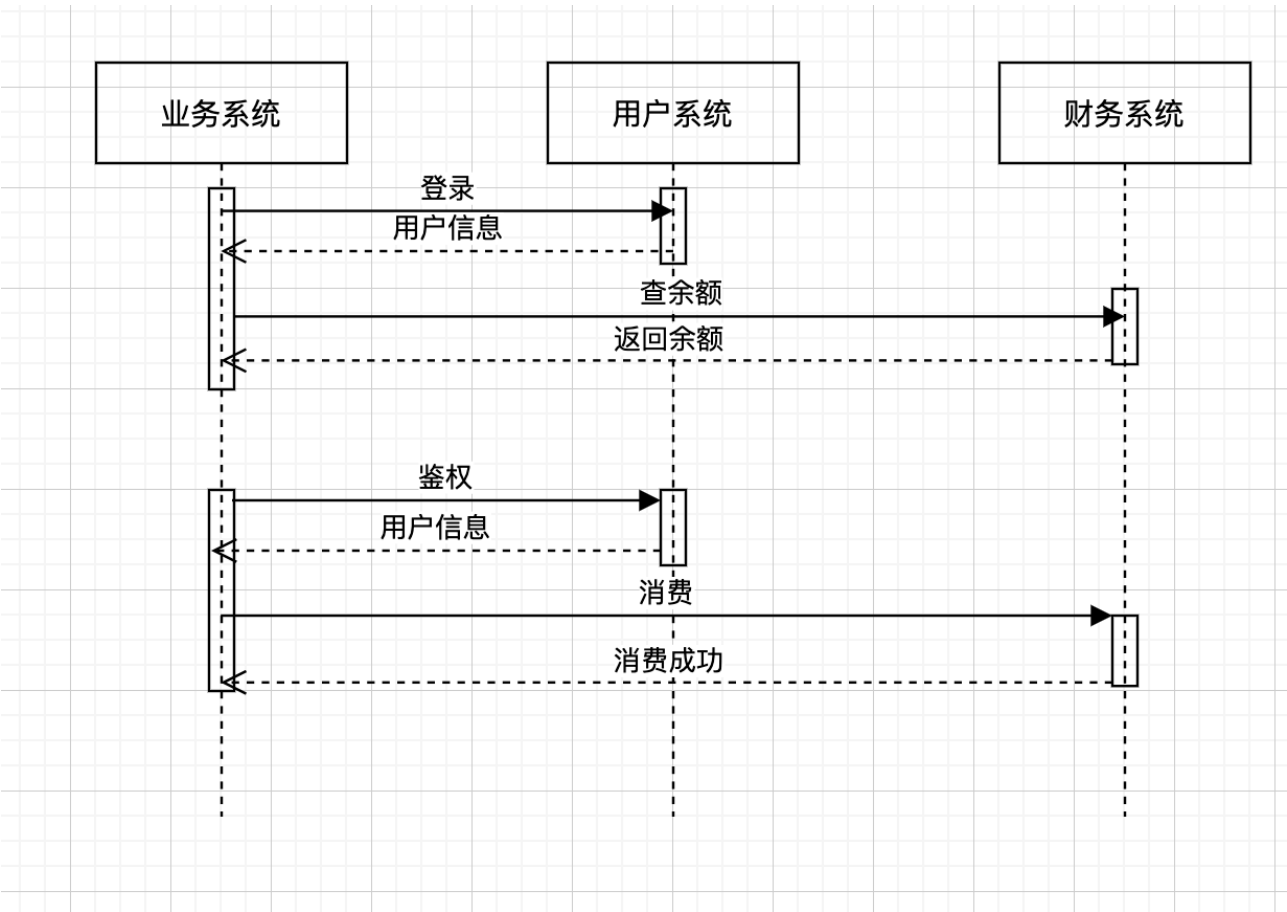
如上图所示，系统内部有三个子系统，其中业务系统部署2台服务器，负责和用户PC以及食堂收费终端交互，提供查询服务以及收费扣款服务。业务系统依赖内部的用户系统对用户进行管理以及登陆鉴权，并依赖财务系统进行费用结算。

财务系统部署2台服务器，提供费用结算的业务逻辑，并负责和第三方供应商，如银行，提供的支付服务进行交互，完成费用的充值结算等一系列操作。

用户系统部署2台服务器，负责用户的登录，鉴权等操作。

存储系统部署3台服务器，为以上三个系统提供状态，数据的存储服务。

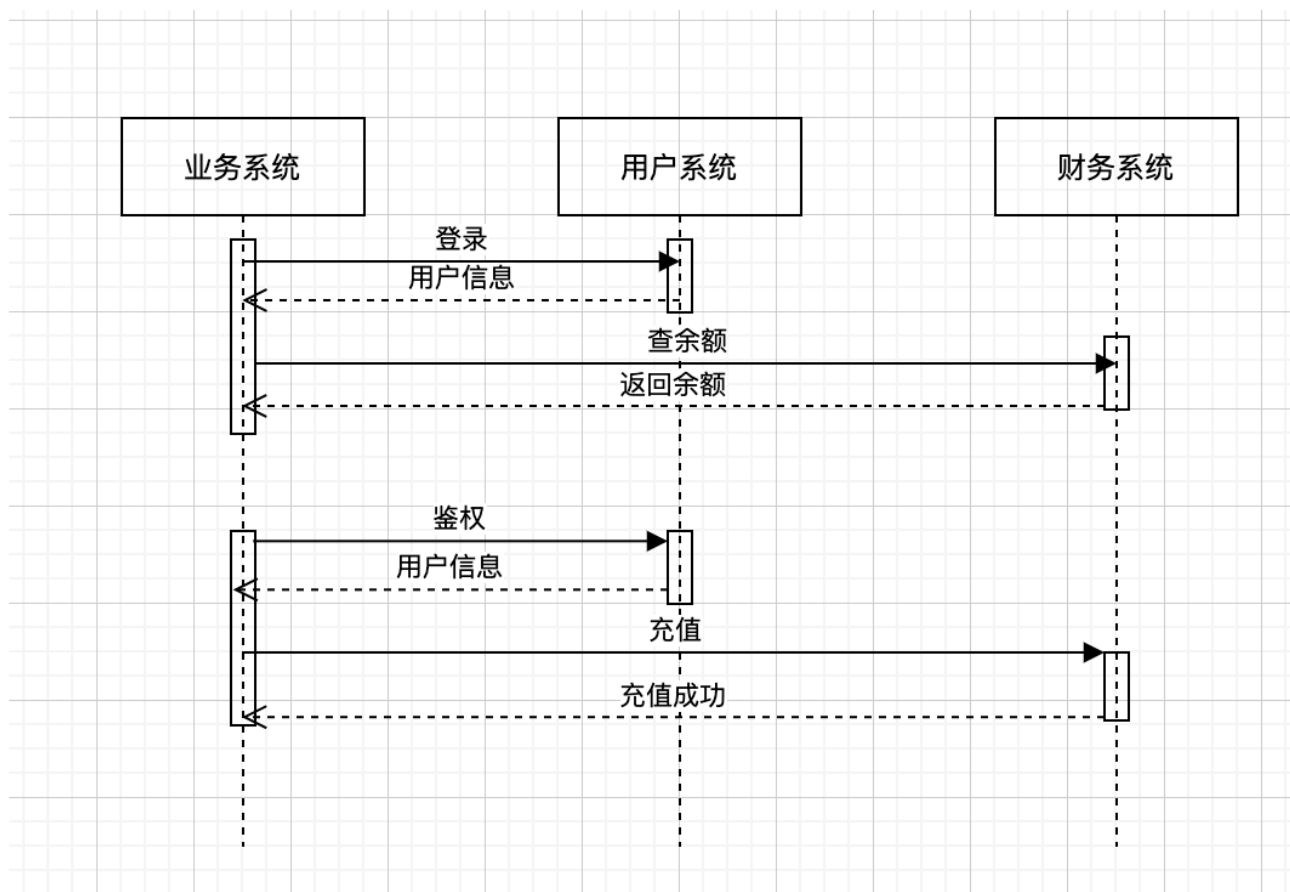
2.2 消费场景系统时序图



如上图所示，是一个消费的流程，用户刷卡时会产生登录动作，业务系统收到请求后发请求给用户系统进行鉴权登录，得到用户的信息，然后再带着用户信息查财务系统，得到用户的余额。

之后收银员操作收费这个动作，业务系统收到请求后，先拿着token去用户系统解析出用户信息，然后带着用户信息给财务系统发消费请求，直到财务返回消费成功的讯号。

2.3 充值场景的子系统时序图



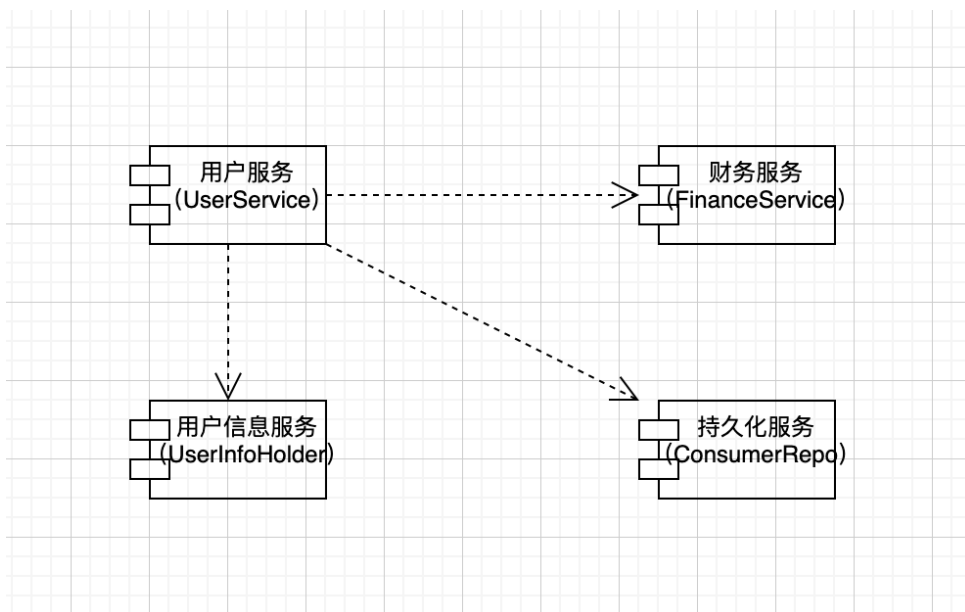
如上图所示，充值场景和消费场景非常类似，只不过请求大部分是用户终端发出而不是收费终端。收到用户终端的请求以后一样是先进行登录并查看余额，之后用户发起充值请求时，同样通过鉴权拿到用户信息，并通知财务系统去进行充值。其中，充值的过程会调用第三方支付服务提供者进行结算。

3. 业务系统设计

业务系统是负责与用户完成交互，并处理复杂业务逻辑的子系统。

3.1 业务系统组件设计

这个子系统由UserInfoHolder, FinanceService, UserService, ConsumerRepo等组件构成：



如上图所示：

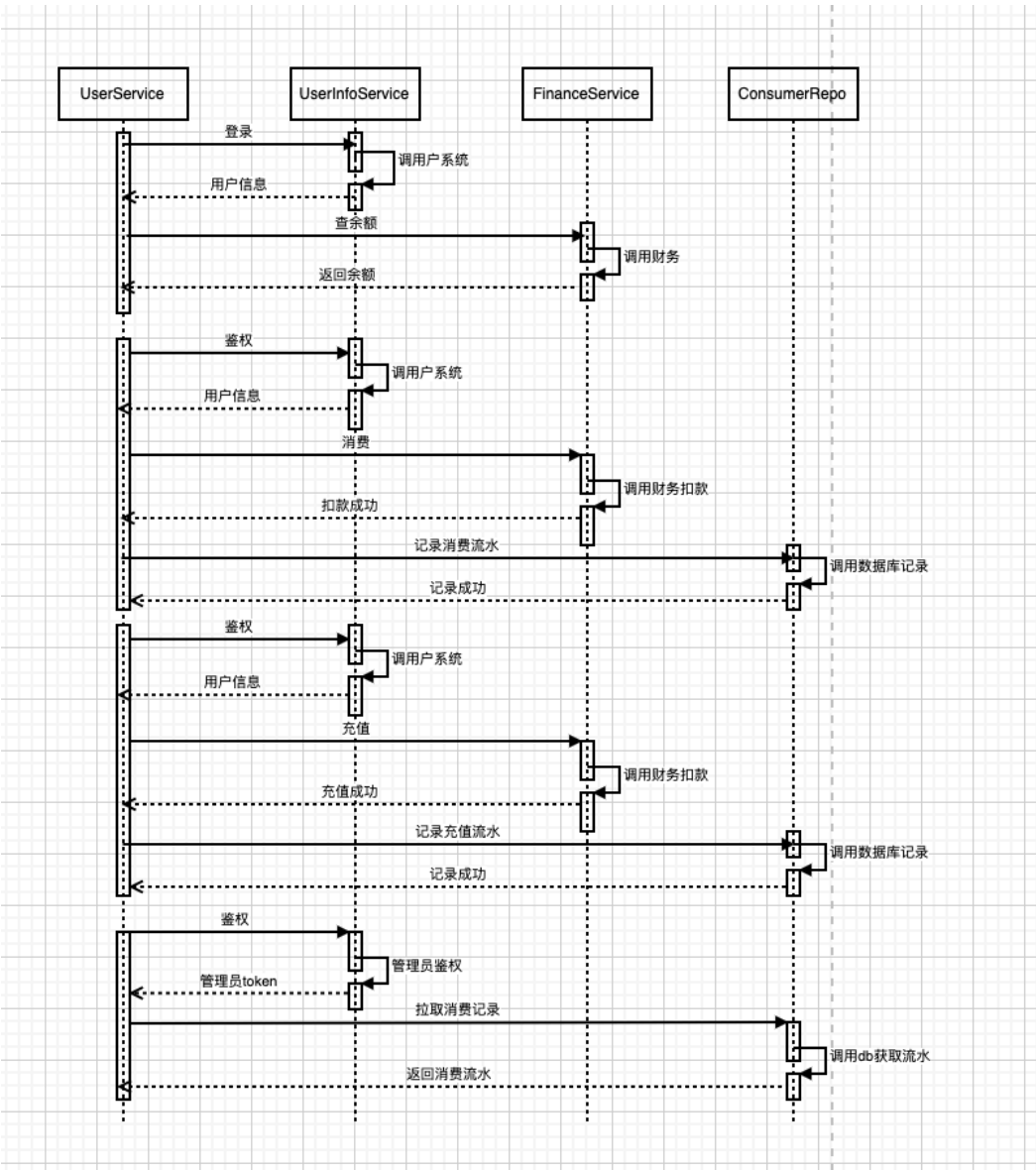
用户服务主要是负责和用户发生交互，并且完成用户的消费，查询，充值等行为动作，而其他服务则为用户服务提供支持。

财务服务负责和财务系统发生RPC交互，传递查余额，扣款，充值等操作。

用户信息服务主要是负责用户信息的维护，在用户服务收到请求的时候都要先调用用户信息服务获取用户信息才能进行下一步行动。用户信息服务本身通过RPC服务从远程的用户系统中获取所需要的用户信息。

持久化服务主要负责和DB发生交互，记录用户的消费，充值流水，以便于日后供管理员监督和查看。

3.2 消费，充值，查询，管理等在业务系统组件的时序图



如上图所示为业务系统的组件时序图。所有的请求进来都需要调用UserInfoService进行鉴权，确定身份以后，再去访问财务系统或者存储系统获取响应的信息，或者进行消费，充值等行为。