# 企业资产管理模块设计

## 模块分析图

### 总流程图

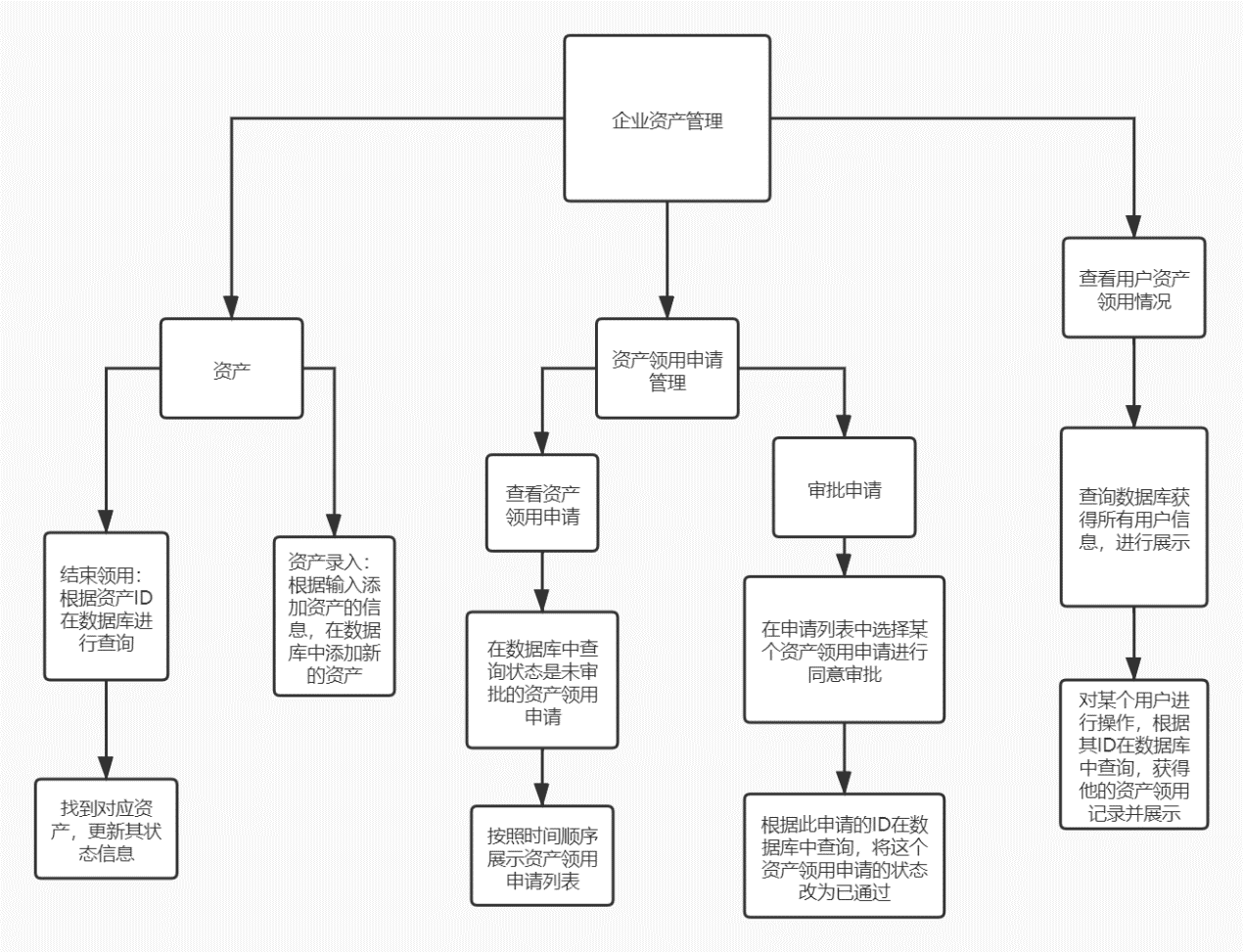


图1 企业资产管理模块流程图

## 功能设计说明

### 企业资产管理模块

本模块主要涉及资产信息和状态的管理工作，包括资产的增添、使用申请的审批和结束、资产信息和状态的查看和更新以及对用户信息和领用历史的查看，主要分为资产录入模块、使用审批模块、资产查看模块、用户管理模块和资产领用结束模块等子模块。

#### 子模块1：资产录入模块

##### 设计图

本模块的处理过程如图所示：

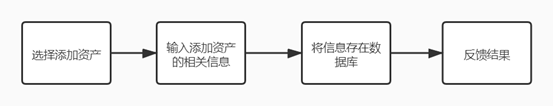


图2 资产录入模块流程图

##### 功能描述

本模块实现的是资产录入功能。此项功能由管理员进行操作。管理员输入要添加的资产的相关信息，包括名称、品牌、型号、规格、数量，输入信息之后进行资产添加。

##### 输入数据

管理员选择添加资产之后，填写资产的相关信息，相关信息有名称、品牌、型号、规格、数量，这些信息作为输入。

##### 输出数据

在填写资产的名称、品牌、型号、规格、数量等相关信息之后，如果输入信息符合规定，就将资产信息添加到数据库。

##### 业务算法和流程

由管理员输入要添加资产的信息，获取输入的信息之后，在数据库中添加新的资产；新添加的资产的各项属性值是对应的各个输入信息。

##### 数据设计

private Integer pId：资产ID。

private String pName：资产名称。

private String pBrand：资产品牌。

private String pModel：资产型号。

private String pSpec：资产规格。

private Date pTime：资产采购日期。

private User user：资产当前的使用用户。

##### 源程序文件说明

ManagerController.java：管理员管理的Controller。

JsonProperty.java：前台传来的Property数据的封装类。

PropertyManageService.java：资产管理的接口。

PropertyManageServiceImpl.java：资产管理的Service层接口实现。

Property.java：资产的类。

##### 函数说明

1. ManagerController.addProperty：调用下列函数2、3、4、5、6、7、8，返回资产添加的结果。
2. JsonProperty.toMap：获得输入的资产信息，返回的是资产的名称、品牌、型号、规格、数量。
3. PropertyManageService.addProperty：添加资产，返回添加的结果；如果添加成功，返回值为true，否则为false
4. Property.setpName：设置添加资产的名称。
5. Property.setpBrand：设置添加资产的品牌。
6. Property.setpModel：设置添加资产的型号。
7. Property.setpSpec：设置添加资产的规格。
8. Property.setpTime：设置资产添加的日期。

#### 子模块2：使用审批模块

##### 设计图

本模块的处理过程如图所示：

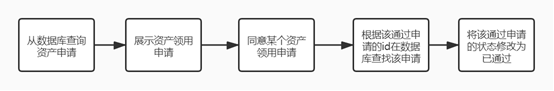


图3 使用审批模块流程图

##### 功能描述

本模块实现的是资产申请审批功能。此项功能由管理员进行操作。管理员可以查看资产申请列表，然后对申请进行操作，可以选择同意某个资产领用申请。

##### 输入数据

进入使用审批界面，查看资产领用申请列表，资产领用申请是按照时间顺序排列；同时，对资产领用申请进行操作，同意某个资产领用申请。

##### 输出数据

在管理员同意某个资产领用申请之后，该项资产领用申请的状态会改为已通过，在使用审批界面不会显示已经通过的资产领用申请。

##### 业务算法和流程

首先根据资产领用申请的状态在数据库中进行查询，然后再展示状态是未审批的资产领用申请。管理员选择同意某个资产领用申请之后，在数据库中查找此项资产领用申请，然后将这个资产领用申请的状态改为已通过。

##### 数据设计

private Integer aId：申请单的ID。

private User user：使用的用户。

private Property property：领用的资产。

private Manager manager：审批的管理员。

private Date beginTime：开始时间。

private Date reviewTime：审批时间。

private Date endTime：结束时间。

private Integer aStatus：申请单状态。

private Integer operation：管理员的操作。

##### 源程序文件说明

ManagerController.java：管理员管理的Controller。

ApplicationManageService.java：申请单管理接口。

ApplicationManageServiceImpl.java：申请单管理的Service层接口实现。

ApplicationRepo.java：访问Application的JPA接口。

Application.java：资产领用申请的类。

##### 函数说明

* 1. ManagerController.getApplicationManage：调用下列的函数2、4、5，获取资产领用申请信息，分页展示申请列表。
  2. ApplicationManageService.findByPage：分页查询，调用函数3，返回资产领用申请列表。
  3. ApplicationRepo.findAllUnChecked：查询所有未审批的资产领用申请。
  4. ApplicationManageService.getNumOfPageByPageAndProperty：通过页面大小和资产来获取页面的数量。
  5. ApplicationManageService.findByPageAndProperty：通过资产进行分页查询。
  6. ManagerController.reviewApplication：调用下列函数7、8、9、10、11，完成审资产领用申请申的审批，改变某个申请的状态。
  7. ApplicationManageService.updateApplication：更新申请的状态，如果管理员拒绝申请，返回true，管理员同意则返回false。
  8. ApplicationRepo.findByaId：通过某个申请ID来查询资产领用申请。
  9. Application.setReviewTime：设置某个资产领用申请审批的时间。
  10. Application.setManager：设置审批某个资产领用申请的管理员。
  11. Application.setaStatus：设置某个资产领用申请的状态。

#### 子模块3：资产查看模块

##### 设计图

本模块的处理过程如图所示：

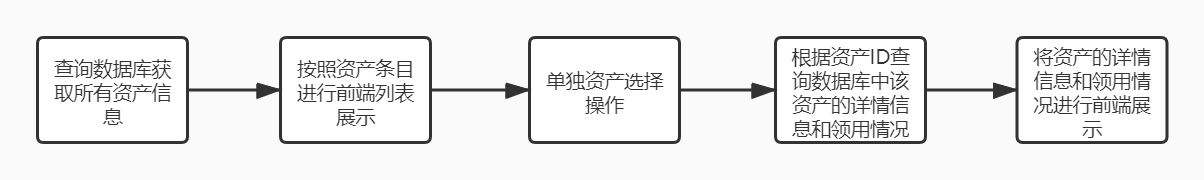


图4 资产查看模块流程图

##### 功能描述

本模块通过获取数据库中所有资产信息，以列表形式进行展现所有资产条目，并可进一步查看单独资产的编号、名称、品牌等有关资产详情和资产的领用历史。

##### 输入数据

数据库中所有资产信息，包括资产的编号、设备名称、品牌、型号、规格、采购日期、领用情况等相关数据。

##### 输出数据

所有资产条目的相关信息的列表分页展示以及单独资产的详情信息和领用情况展示。

##### 业务算法和流程

通过查询数据库获取所有资产的相关信息，传递到前端进行列表分页展示；通过选择特定资产，获取对应ID进行数据库查询并前端展示，实现单独资产的详情信息和领用情况的查看操作。

##### 数据设计

properties：List<Property>对象，个体为组成一页的property对象数组。

applications：List<Application>，个体为针对单独资产组成一页的application对象数组。

另外还有id，name等一系列资产相关的属性参数和curPage，sumPage等有关分页的数据。

##### 源程序文件说明

PrepertyRepo.java：访问Property的JPA接口。

PropertyManageService.java：资产管理的Service接口。

PropertyManageServiceImpl.java：资产管理的Service接口的实现类。

ApplicationManageService.java：申请单管理的Service接口。

ApplicationManageServiceImpl.java：申请单管理的Service接口的实现类。

ManagerfController.java：管理员管理的Controller。

propertyManage.jsp：资产管理的jsp页面。

propertyDetail.jsp：单独资产详情的jsp页面。

PMListTag.java：PropertyManage中列表的SimpleTag处理器类。

PDListTag.java：PropertyDetail中列表的SimpleTag处理类。

##### 函数说明

1. PropertyManageService.getNumOfPage：获取所有资产的页面数量，10个为一页。
2. PropertyManageService.findByPage：分页查询，按页返回对应的10个资产property对象。
3. ManagerfController.propertyManage：返回资产详情的视图，调用PropertyManageService的getNumOfPage和findByPage方法得到所有资产的列表对象properties以及分页的相关参数，传递给前端propertyManage页面。
4. PMListTag.doTag：PropertyManage中列表的SimpleTag的doTag方法覆写，获取参数所有资产properties，实现每个资产property的属性展示。
5. ApplicationManageService.getNumOfPageByPageAndProperty：通过页面大小和资产来获取页面的数量。
6. ApplicationManageService.findByPageAndProperty：通过资产进行分页查询。
7. PropertyManageService.findBypId：通过id查找资产。
8. ManagerfController.getPropertyDetail：获取资产详情，调用PropertyManageService的findBypId和ApplicationManageService的getNumOfPageByPageAndProperty、findByPageAndProperty方法得到对应资产的详情信息和使用历史信息以及分页的相关参数，传递给前端propertyDetail页面。
9. PDListTag.doTag：PropertyDetail中列表的SimpleTag的doTag方法覆写，获取参数所有申请applications，实现单独资产使用历史（领用历史）的信息展示。

#### 子模块4：用户查看模块

##### 设计图

此模块的功能结构如下所示：

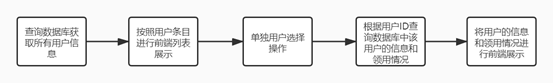


图5 用户查看模块流程图

##### 功能描述

本模块通过获取数据库中所有用户信息，以列表形式进行展现所有用户条目，并可进一步查看单独用户的用户名、邮箱、账户状态等用户详情信息及其资产领用历史。

##### 输入数据

数据库中所有用户信息，包括用户的用户名、邮箱、账户状态等用户详情信息及其资产领用历史等相关数据。

##### 输出数据

所有用户条目的相关信息的列表分页展示以及单独用户的详情信息和资产领用历史展示。

##### 业务算法和流程

通过查询数据库获取所有用户的相关信息，传递到前端进行列表分页展示；通过选择特定用户，获取对应ID进行数据库查询并前端展示，实现单独用户的详情信息和资产领用历史的查看操作。

##### 数据设计

users：List<User>列表对象，个体为组成一页的user对象数组。

applications：List<Application>，个体为针对单独用户组成一页的application对象数组。

另外还有id，name等一系列用户相关的属性参数和curPage，sumPage等有关分页的数据。

##### 源程序文件说明

UserRepo.java：访问User的JPA接口。

UserManageService.java：用户管理的Service接口。

UserManageServiceImpl.java：用户管理的Service接口的实现类。

ApplicationManageService.java：申请单管理的Service接口。

ApplicationManageServiceImpl.java：申请单管理的Service接口的实现类。

ManagerfController.java：管理员管理的Controller。

userManage.jsp：用户管理的jsp页面。

userDetail.jsp：单独用户详情的jsp页面。

UMListTag.java：UserManage中列表的SimpleTag处理器类。

UDListTag.java：UserDetail中列表的SimpleTag处理类。

##### 函数说明

1. UserManageService.getNumOfPage：获取所有用户的页面数量，10个为一页。
2. UserManageService.findByPage：分页查询，按页返回对应的10个用户user对象。
3. ManagerfController.userManage：返回用户管理对应分页下的信息，调用UserManageService的getNumOfPage和findByPage方法得到所有用户的列表对象users以及分页的相关参数，传递给前端userManage页面。
4. UMListTag.doTag：UserManage中列表的SimpleTag的doTag方法覆写，获取参数所有用户users，实现每个用户user的属性展示。
5. UserManageService.findByuId：通过id查找用户。
6. ApplicationManageService.getNumOfPageAndUser：通过页面大小和用户来获取页面的数量。
7. ApplicationManageService.findByPageAndUser：通过用户进行分页查询。
8. ManagerfController.getUserDetail：获取用户管理下的用户详情的视图，调用UserManageService的findBypId和ApplicationManageService的getNumOfPageAndUser、findByPageAndUser方法得到对应用户的属性信息和使用历史信息以及分页的相关参数，传递给前端userDetail页面。
9. UDListTag.doTag：UserDetail中列表的SimpleTag的doTag方法覆写，获取参数所有申请applications，实现单独用户使用历史（领用历史）的信息展示。

#### 子模块5：资产领用结束模块

##### 设计图

此模块的功能结构如下所示：

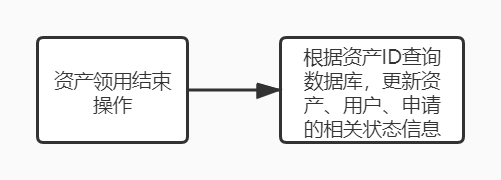


图6 资产领用结束模块流程图

##### 功能描述

本模块针对处于使用状态的资产，进行结束申请、结束领用的信息更新，并更新资产状态。

##### 输入数据

需要进行结束领用操作的资产ID。

##### 输出数据

数据库和前端有关该资产的状态信息、申请条目信息和相关用户状态信息的更新。

##### 业务算法和流程

通过选择特定资产，获取对应ID进行数据库查询，获得对应的申请条目，继而获得对应的申请用户，最后进行数据库该资产的状态信息、申请条目信息和相关用户状态信息的数据更新。

##### 数据设计

无。

##### 源程序文件说明

PrepertyRepo.java：访问Property的JPA接口。

PropertyManageService.java：资产管理的Service接口。

PropertyManageServiceImpl.java：资产管理的Service接口的实现类。

ApplicationManageService.java：申请单管理的Service接口。

ApplicationManageServiceImpl.java：申请单管理的Service接口的实现类。

ManagerfController.java：管理员管理的Controller。

propertyDetail.jsp：单独资产详情的jsp页面。

##### 函数说明

1. PropertyManageService.findBypId：通过id查找资产。
2. PropertyManageService.endApplicationBypId：通过资产id查询对应资产，结束资产被领用的状态。
3. ApplicationManageService.findByproperty\_pId\_on：通过资产id查找现行的通过申请。
4. ApplicationManageService.endApplicationByaId：通过申请id查找对应申请，将其状态设置为结束。
5. ManagerfController.delApplication：结束领用，通过前端propertyDetail传回的对应资产的结束领用触发事件和对应资产id，调用PropertyManageService的findBypId和endApplicationBypId方法，改变对应资产的被领用状态；调用ApplicationManageService的findByproperty\_pId\_on 和endApplicationByaId方法，结束对应的申请。