



山东建筑大学

本科毕业设计说明书

题 目： 军械存储保障系统的设计与实现

院（部）： 计算机科学与技术学院

专 业： 网络工程

班 级： 152

姓 名： 王斌

学 号： 201511102016

指导教师： 刘刚、秦松

完成日期： 2019 年 5 月 20 日

目 录

| | |
|--------------------|-----|
| 摘 要 | III |
| ABSTARCT | IV |
| 1 前 言 | |
| 1.1 研究背景..... | 1 |
| 1.2 研究内容..... | 1 |
| 1.3 研究意义..... | 1 |
| 2 需求分析 | |
| 2.1 需求性背景分析 | 2 |
| 2.2 功能性需求分析 | 2 |
| 2.3 系统性用户分析 | 3 |
| 2.4 安全性需求分析 | 4 |
| 2.5 可靠性需求分析 | 4 |
| 2.6 可行性需求分析 | 4 |
| 2.7 运行环境的分析 | 4 |
| 2.7 开发技术和语言介绍..... | 5 |
| 3 概要设计 | |
| 3.1 系统分析..... | 6 |
| 3.2 系统功能结构..... | 10 |
| 3.3 数据库设计 | 13 |
| 4 详细设计及实现 | |
| 4.1 登录..... | 20 |
| 4.2 首页信息展示..... | 21 |
| 4.3 军械管理界面..... | 23 |

| | |
|-------------------|----|
| 4.4 入库管理 | 26 |
| 4.5 借用管理 | 27 |
| 4.6 维护管理 | 29 |
| 4.7 报废管理 | 32 |
| 4.8 巡查管理 | 34 |
| 4.9 基础管理 | 36 |
| 5 系统测试 | |
| 5.1 测试目的 | 41 |
| 5.2 测试用例 | 41 |
| 6 结论 | 46 |
| 谢 辞 | 47 |
| 参考文献 | 48 |

摘 要

军械存储保障管理系统是在各国家应用及其广泛的军械系统，但是随着信息化技术的发展，老旧的军械系统已经不能满足现代化军队的需求。基于这一现象，本毕业设计模拟实现了新一代的军械存储保障管理系统。在以往的系统中，军械信息采集以及军械出入库管理都存在着较大的瑕疵与风险，往往存在着军械出库与入库信息不符，最后导致军械信息丢失的现象，引发严重问题。所以，本系统开发了一个具有良好的安全性与界面友好性、军械信息记录严格、军械出入库信息公开透明、军械维护和报废信息有源可查以及部队人员角色权限严格限制的一个适用基层部队装备管理的综合性军械存储保障管理系统。本系统采用 MVC 模式使用功能以及安全性都比较全面的 SQLite 数据库系统和微软的 Microsoft Visual Studio，基于.NET 平台进行程序设计。

关键词：军械存储；军械管理；MVC；SQLite；.NET

ABSTARCT

Ordnance storage and support management system is an ordnance system widely used in various countries, but with the development of information technology, the old ordnance system can not meet the needs of modern military. Based on this phenomenon, the simulation of the graduation project to achieve a new generation of ordnance storage security management system. In the previous system, the ordnance information collection and ordnance storage and storage management have great defects and risks, and there is often a discrepancy between ordnance storage and storage information, resulting in the ordnance information loss phenomenon, causing serious problems. So, the security of this system to develop a good and friendly interface, weapons loading and unloading information open and transparent information record strictly, weapons, weapons maintenance and scrap can check information of active and forces role authorization strictly an available grassroots units and equipment management comprehensive ordnance store security management system. This MVC's system USES SQLite database system with comprehensive functions and security, and Microsoft Visual Studio, based on.net platform for program design.

Keywords: Ordnance storage, Ordnance management, MVC, SQLite, NET

1 前言

1.1 研究背景

伴随着现代化信息技术的发展，信息系统越来越应用到广泛的领域，并且在一些极具机密性的领域中更是占有者不可替代的地位。在原来的存储管理系统中，层层审批，以及数据的纸质化记录和登记已经严重的拉低了军械管理的效率。为了充分实现信息化管理让部队进行飞跃性发展，设计一个信息化的武器存储管理系统是十分具有必要性的。

1.2 研究内容

本项目设计开发了一个基于网络环境的军械装备存储保障系统。该系统为部队人员提供了便捷的武器出入库管理、武器维护、武器报废以及记录查询平台。部门操作员只要在用户激活的情况下就可以进行武器的出入库申请，仓库管理人员可以随时对武器进行出入库管理，维修人员以及采购员相应的对武器进行维护和采购，而高级部门管理人员可以对各种记录进行巡视查询。各级人员按照角色进行处理，拥有着不同的权限，根据此权限进行功能屏蔽。有效的避免了多种用户针对统一操作平台进行错误操作而导致了军械管理存储的混乱与失误，方便了各级人员对军械装备存储保障系统的操作。针对军械存储管理出入库的业务流程接合科学管理的理论，本文开发了一个基于.NET 开发平台的军械装备存储保障系统，结合 C#语言，以 SQLite 作为后台数据库。

1.3 研究意义

部队长期以来在已经落后的军械装备存储保障系统上面进行武器出入库管理，操作审批流程比较繁琐，界面不友好。传统的系统存在着的诸多的缺点限制着军队现代化的发展。本文基于 layuiUI 进行界面规划设计，使用 c#，实现了具有检索功能的快捷信息管理系统，是更为简单的高效率的军械装备存储保障系统。

2 需求分析

2.1 需求性背景分析

伴随着信息化、数字化、网络化的蓬勃发展，军队的武器存储管理也越来越繁重以及严格，原来的军械装备存储保障系统已经不能完全满足相关人员使用的需要。最大化利用现有技术以及资源打造更具有现代化气息的管理系统来提高部队效率是部队目前亟待解决的问题，因此实现军械装备存储保障系统是势在必行的。

2.2 功能性需求分析

军械装备存储保障系统的总体开发任务是实现部队军械装备的存储管理的规范化、信息化以及系统化。为军队各级部门人员提供统一的平台，在符合规范的情况下，在该系统进行武器装备的出入库管理、武器装备的维修报废以及各种记录的查询。各级人员按照自己的权限进行相应的操作之后会有相应的提示。系统为部队的各级人员解决有关武器出入库维修报废等繁琐工作，减少各级人员的工作量，提高军队现代化管理的任务。

本系统的主要功能：

（1） 装备入库

实现武器装备信息的录入、入库数量、维护周期、入库仓库的管理，以及增删改查

（2） 装备出库

装备借出数量，可借数量，借出申请的管理，包括增删改查

（3） 装备维护

装备维修周期的显示，维修记录的显示，维修状态的显示，包括增删改查

（4） 装备报废

装备的报废记录、报废数量以及报废仓库，包括增删改查

(5) 基础管理

用户角色的管理、仓库的基本管理、日常巡视记录的管理，包括增删改查

2.3 系统性用户分析

本系统角色一共分为七类

(1) 采购员

进入军械装备存储保障系统之后，在军械入库采购中可以进行军械入库的军械添加的填写，查看一些本系统的基础信息

(2) 部门操作员

部门管理人员登录之后，除了可以查看一些武器系统的基础信息之外，主要是对武器演习出库申请进行操作，选择武器种类以及仓库，填写配送人员信息以及联系方式

(3) 仓库管理员

仓库管理人员是除了操作员之外的第二大用户。除了信息的查看之外，还可以进行武器装备信息的录入，出入库管理的拒绝与同意。

(4) 维修人员

负责对在维修周期中的武器进行维修，并且存有维修记录

(5) 巡查人员

不定期对各种武器进行巡查记录、并及时把出现的问题反馈以及在系统中记录问题的实现与反馈

(6) 部门领导

可不定期对仓库进行巡查，各种维护记录、报废记录以及军械出入库记录进行检查与核对，防止出现武器纰漏的情况

(7) 系统管理员

系统管理员负责整个系统的基础管理，类似于用户添加，角色管理、仓库管理等等，拥有系统的所有权限，负责系统的安全以及稳定

2.4 安全性需求分析

鉴于军械装备存储保障系统有着非常庞大的数据量,因此对这些军械武器数据进行查询时以及添加时要保持着极高的效率,还要保证查询的准确性以及完整性。对于本军械装备存储保障系统而言,不仅仅是要修改原始记录,还要在一定程度上保持军械数据库中数据的完整性,防止军械数据的丢失,要求具有完整权限分配,防止军械系统被恶意攻击。

2.5 可靠性需求分析

可靠性从安全性、事务性和稳定性进行分析权衡,安全性方面,由于本系统是围绕部队军械装备保障系统进行设计的,是面向武装力量的,所以对于安全性和保密性的要求相对来说比较高。对于事务性而言,利用 SQLite 本身的处理机制进行规划与保障。而对于稳定性来说,系统越复杂,信息量越大,出现错误的情况就会更高,因此要判断出错误出现率以及更正效率以及时间,再发生故障的状态下的可恢复性以及可预见性是怎样的。

2.6 可行性需求分析

本军械装备存储保障系统安装实行之后,会提高普通业务流程的信息化管理的效率,在一定程度上节约了管理军械的成本,节约了开发资金,降低了各级部门对于军械管理的成本与投入,所以在资金经济方面,本系统是可行的。开发本军械装备存储保障系统,需要有对军械进行出入库、维护报废以及记录查询等等一系列的功能的实现,通俗的来讲就是对数据库的增删改查,并且要用到可视化开发的高级语言来保障系统的稳定性以及友好的界面来增加用户体验得舒适度。对于用户来说,只需要安装好浏览器即可使用本系统,无需其他的配套,减少维护成本以及开发成本,提高部队中各部门的办事效率。所以,本军械装备存储保障系统在技术上也是可行的。

2.7 运行环境的分析

本系统在任何安装浏览器的电脑都可以运行。

2.8 开发技术和语言介绍

2.8.1 LayUI 框架简介

是一款采用自身模块规范编写的前端 UI 框架，遵循原生 HTML/CSS/JS 的书写与组织形式，门槛极低，拿来即用。其外在简洁，却又不失饱满的内在，体积轻盈，组件丰盈，从核心代码到 API 的每一处细节都经过精心雕琢，非常适合界面的快速开发。layUI 首个版本发布于 2016 年秋，她区别于那些基于 MVVM 底层的 UI 框架，却并非逆道而行，而是信奉返璞归真之道。准确地说，她更多是为服务端程序员量身定做，无需涉足各种前端工具的复杂配置，只需面对浏览器本身。

2.8.2 .NET 平台与语言简介

.NET 是 Microsoft XML Web services 平台。XML Web services 允许应用程序通过 Internet 进行通讯和共享数据，而不管所采用的是哪种操作系统、设备或编程语言。Microsoft .NET 平台提供创建 XML Web services 并将这些服务集成在一起之所需。对个人用户的好处是无缝的、吸引人的体验。C#语言是一种面向对象的、运行于.NET Framework 和.NET Core(完全开源，跨平台)之上的高级程序设计语言，在继承 C 和 C++强大功能的同时去掉了一些它们的复杂特性。

2.8.3 SQLite 数据库简介

SQLite，是一款轻型的数据库，是遵守 ACID 的关系型数据库管理系统，它包含在一个相对小的 C 库中。它是 D.RichardHipp 建立的公有领域项目。它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百 K 的内存就够了。它能够支持 Windows/Linux/Unix 等等主流的操作系统，同时能够跟很多程序语言相结合，比如 Tcl、C#、PHP、Java 等，还有 ODBC 接口，同样比起 Mysql、PostgreSQL 这两款开源的世界著名数据库管理系统来讲，它的处理速度比他们都快。

3 概要设计

3.1 系统分析

3.1.1 系统功能

本系统需要满足各级人员的不同操作。按照对于武器操作来划分主要分为以下几大模块：武器出库、武器入库、武器维护、武器报废、巡查记录、仓库管理以及用户管理等等。而针对于这几大模块，下属都会涵盖有以下几个小功能的实现：

(1) 记录添加

显示对于武器操作之后的各个记录，以及相应的操作数据，例如武器名称，武器批次，维修时间，维修周期，所属仓库等等。

(2) 记录查询

通过输入武器的关键词或者限定日期之后，可以查找到需要的武器操作记录，而不需要逐个比对。

(3) 记录删除

在符合操作条件的情况下，对武器记录进行删除的操作，删除数据库中的记录，防止出现误删错删的情形。

(4) 记录修改

针对武器进行记录的添加之后发现武器记录发生了错误，或者与预期的信息不符合，可以在未入库的情况下对武器信息进行修改。

3.1.2 系统主要流程

(1) 装备借出

部门操作员登录之后，进入借用管理下借用信息页面，点击添加按钮，进行装备借用信息的填写，军械名称、借用仓库、军械批次、借用数量、借用人、提取人姓名，联系方式等等，然后点击提交按钮，仓库管理人员会进行判断，是否同意借出，如果仓库管理人员否定了本次军械使用的借出，那么本次操作就会无

效，但是仍然会有借出记录显示已拒绝状态，否则会借出成功，显示成功借出，并会修改可借数量等等。

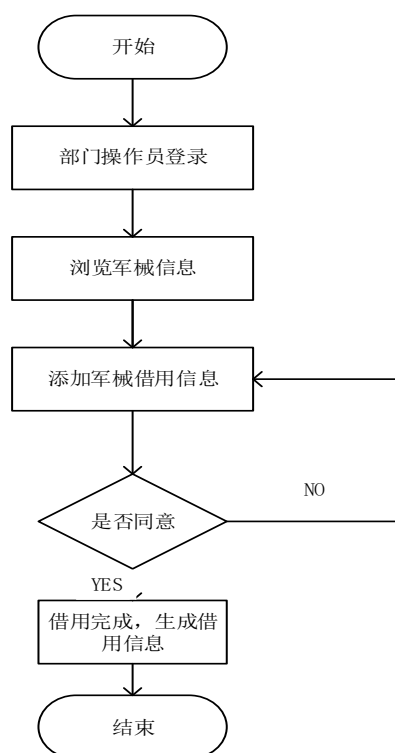


图 3.1 装备借出流程图

(2) 装备维护

装备维护人员登录，进入维护管理页面、页面会根据装备信息添加时的维修周期，进行需要维修装备的记录显示，并且只有当军械没有借出数量的时候才会显示需要维修的军械。此时，维修人员选择需要维修的装备、点击维修按钮进行装备维修，系统会修改维修状态显示在维修当中，维修完成后，准备会回库，并增加一条维修记录。

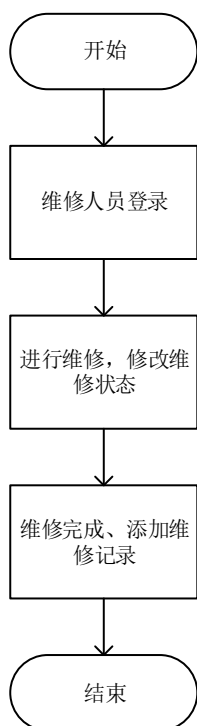


图 3.2 装备维护流程图

(3) 装备添加

管理员登录之后，进入军械管理界面，在军械信息的页面当中点击添加按钮，进行军械装备信息的添加以及删除。在进入入库管理界面下的子界面入库信息，点击添加按钮，选择军械信息、入库仓库、填写军械批次以及军械数量，选择采购员，输入配送人姓名以及联系方式，点击确定按钮，进行入库信息添加。仓库管理员登录之后进行信息的确认是否入库，如果回拒在，则入库信息失败，如果入库，那么会显示入库成功。并且在军械管理的子页面军械数据下面显示出入库的信息。

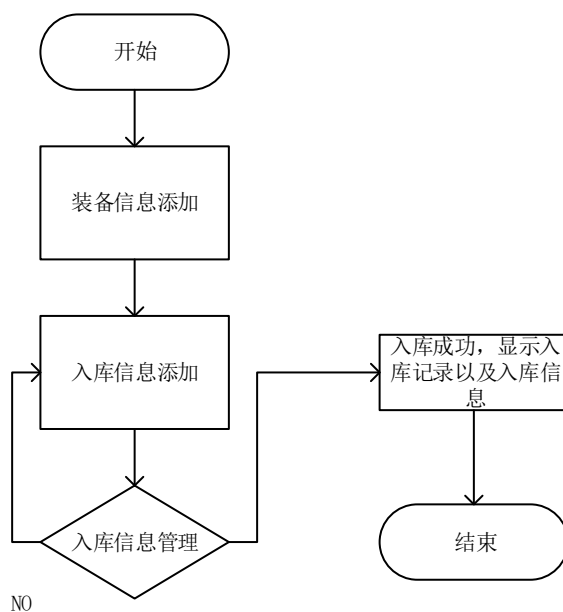


图 3.3 装备添加流程图

(5) 军械报废

仓库管理人员登录之后，进入报废管理下面的子页面报废信息页面，然后点击添加按钮，选择近军械名称、报废仓库、军械批次以及报废数量、点击确定按钮，如果报废失败则需要重新填写报废信息，如果报废成功，那么就会天添加一条报废记录，可借出数量也会进行修改。

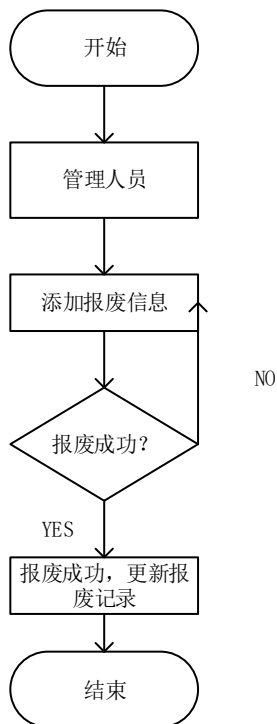


图 3.4 装备报废流程图

(6) 记录查询

领导人员登录之后，进入各个页面，进行军械记录的查询查询装备使用记录、装备维修记录、装备报废记录以及存在的各个部门人员。

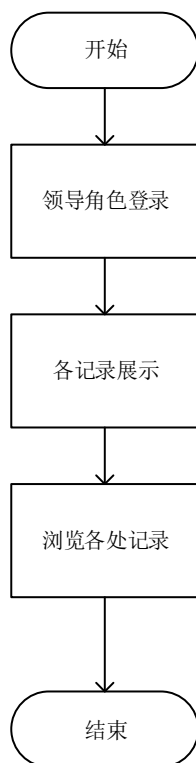


图 3.5 记录查询流程图

3.1.3 系统目标

开发动态的页面，实现页面的动态刷新和异步提交。

实现界面友好性，增加用户体验以及用户好感度。

方便部门各级人员操作，防止出现操作失误，对不同级别人员区别对待。

保证军械装备存储的安全性，防止装备丢失，无记录等。

3.2 系统功能结构

3.2.1 系统时序图

时序图的设计对于系统的各个角色对象和功能有着十分重要的意义，可以理解为输入输出事件的集合。从总体的简单流程上面显示出各个管理逻辑功能的实

现。

(1) 部队人员管理时序图

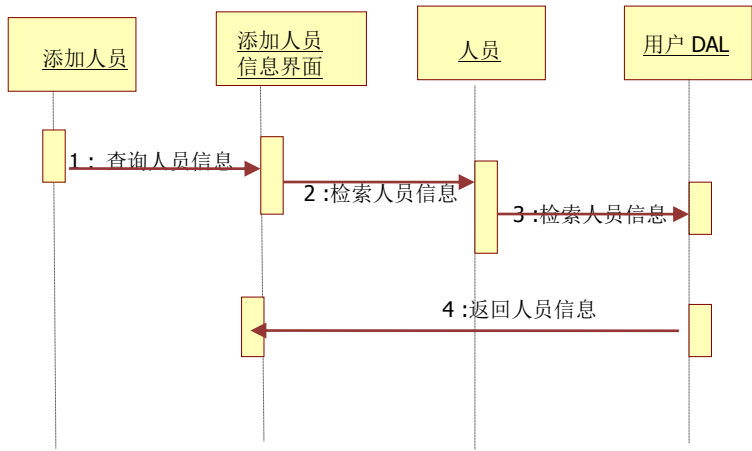


图 3.6 部队人员管理时序图

(2) 军械信息管理时序图

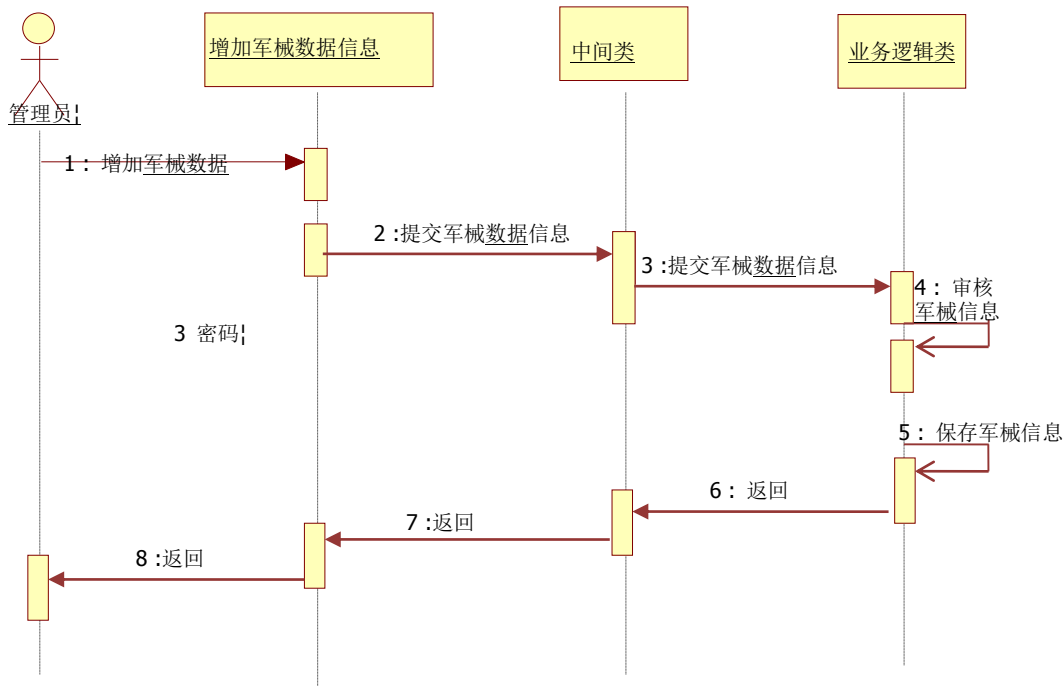


图 3.7 军械信息管理时序图

3.2.2 系统模块设计

表 2.1 模块设计表

| 模块名称 | 功能描述 | 备注 |
|--------|---|--|
| 账号管理模块 | 实现各类用户的（仓库管理员、维修人员、部门管理人员、系统管理员、采购员、系统管理员、领导等）人员信息的添加与删除 | 数据库中默认存储了系统管理员账户，并且系统管理员账户默认拥有所有权限 |
| 角色管理模块 | 显示各个用户的现在状态、以及创建时间 | |
| 仓库管理模块 | 实现各个仓库信息的添加以及删除，并且设定仓库管理人员 | 一个仓库默认只有一个管理人员 |
| 巡查记录模块 | 巡查员不定期巡查、并且将巡查记录记入在库，以便随时查看或者等待部队领导检查 | 巡查记录应由巡查员填写 |
| 军械信息模块 | 现在军械表中添加军械信息，确定是否将军械入库 | 只有在军械表中添加了军械信息之后才能在军械添加模块进行武器的入库添加 |
| 军械添加模块 | 将军械添加到仓库之中，军械是分添加批次、添加仓库以及添加数量的，提交之后会根据操作（拒绝、确认、入库等）进行军械信息的显示 | 军械添加的前提是需要有军鞋的信息在军械库中。被回据的军械信息不能被借出等等。 |

| | | |
|-----------|--|---|
| 军械维护模块 | 军械维护是按照军械入库时存在的维修周期来进行维护的,而维护计划表就是按照维修周期显示需要维护信息,维修时存在维修中以及回库等操作 | 当军械有借出时不能被维修、只有当军械借出数量为 0 的时候才能按照维护计划表进行军械的维修 |
| 军械借出、归还模块 | 军械借出是按照军械可借出数量作为基准的,进行军械借出的时候会按照借出数量以及报废数量等计算出当前可借出的数量。在军械归还时,会重新修改军械的可借出数量以及借出状态。 | 针对于军械出借,可以借给不同的部队,军械借出是由部门管理人员进行操作的 |
| 军械报废模块 | 军械报废模块是仓库人员按照计划进行军械的报废申请,会由部门人员审核之后进行军械的报废 | 军械报废需提交申请,审核批准之后才可以报废,报废数量会影响实际可使用数量 |

3.3 数据库设计

3.3.1 ER 图

通过以上对系统需求的分析,绘制出本系统的总体 E-R 关系图。

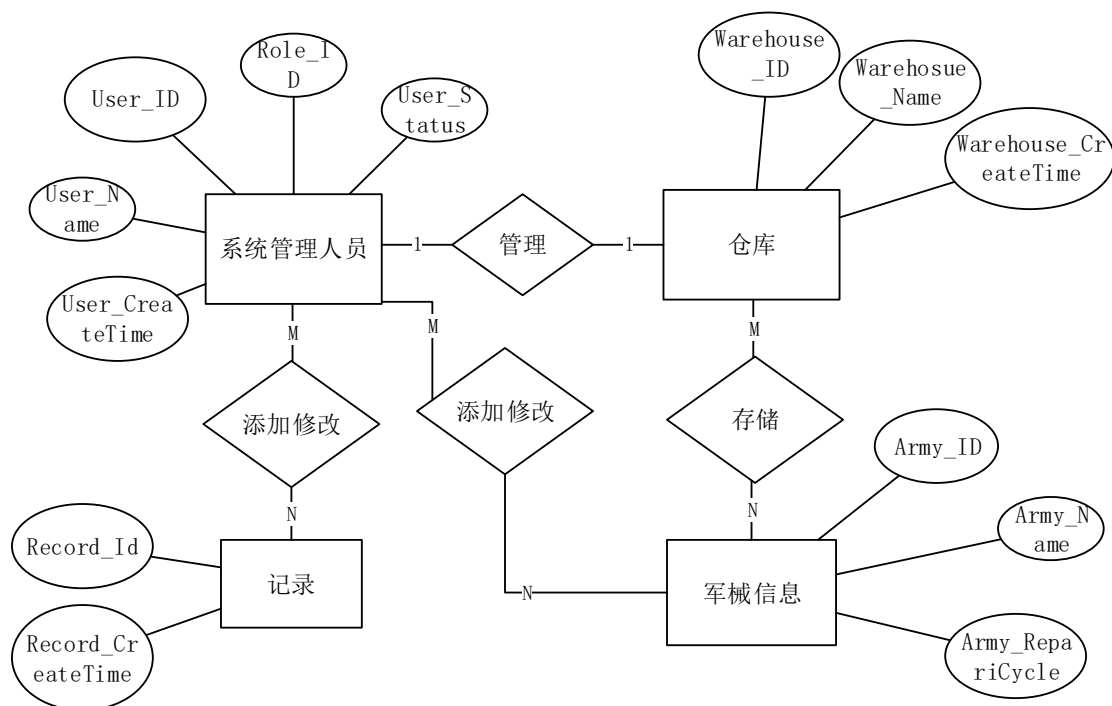


图 3.8 系统 ER 图

3.3.2 表设计

本系统的数据库包含了一下几个数据表：

Ord_ArmyTable 表：军械信息表，用来记录所有军械简要信息，主要包含军械名称，军械所属仓库，以及军械维护周期，在军械添加之初，此表是关键表，只有当此表中包含入库的军械信息，才能进行入库。而军械周期是维护计划表的重要依据，在此基础上查询生成维护计划表。

表 3.1 Ord_ArmyTable

| 字段名称 | 含义 | 类型 | 约束 |
|------------------|-----------|---------------|----|
| Army_ID | 军械 ID | nvarchar(200) | 主键 |
| Army_Name | 军械名称 | nvarchar(50) | 可空 |
| Warehouse_ID | 军械所属仓库 ID | nvarchar(200) | 可空 |
| Army_CreateTime | 军械创建时间 | nvarchar(200) | 可空 |
| Army_RepairCycle | 军械维修周期 | int | 可空 |

Ord_BorrowTable: 军械借用表, 用来记录军械的借用信息。对借出军械 ID、借出用户、借出数量、运送人等等信息进行详细的记录, 这样在出库的时候就从根本上保证了数据的一致性以及军械安全性, 防止归还时借出数量不符或者是借出人信息不符等失误出现。

表 3.2 Ord_BorrowTable

| 字段名称 | 含义 | 类型 | 约束 |
|-----------------------|---------|----------------|-----|
| Borrow_ID | 借出记录 ID | int | 主键 |
| Borrow_ArmyID | 借出军械 ID | nvarchar(200) | 可空 |
| Borrow_WarehouseID | 借出仓库 ID | nvarchar(200) | 可空 |
| Borrow_Batch | 借出军械批次 | nvarchar(50) | 可空 |
| Borrow_UserID | 借出用户 ID | nvarchar(200) | 可空 |
| Borrow_Num | 借出数量 | int | 可空 |
| Borrow_Status | 借出状态 | int | 不可空 |
| Borrow_Time | 借出时间 | nvarchar(200) | 可空 |
| Borrow_ReturnTime | 返还时间 | nvarchar(200) | 可空 |
| Borrow_Remarks | 借出标记 | nvarchar(1000) | 可空 |
| Borrow_TransportName | 运送人姓名 | varchar(50) | 可空 |
| Borrow_TransportPhone | 运送联系方式 | varchar(50) | 可空 |

Ord_MaintainTable: 军械维修表, 用来记录所有的军械维修信息。此表是维修完成之后的维修记录表, 在前面的军械信息表中按照维修周期与军械入库表上次维修时间进行时间差的对比, 生成维修计划表, 维修人员按照此虚拟表的数据进行武器军械的维修, 维修完成后进行回库的操作, 并且系统会自动生成维修记录记录在此表当中。

表 3.3 Ord_MaintainTable

| 字段名称 | 含义 | 类型 | 约束 |
|---------------|--------|--------------|----|
| Maintain_ID | 维修 ID | int | 主键 |
| Maintain_Name | 维修军械名称 | varchar(100) | 可空 |

| | | | |
|------------------------|---------|--------------|----|
| Maintain_Batch | 维修批次 | varchar(100) | 可空 |
| Maintain_WarehouseID | 维修仓库 ID | nvarchar(50) | 可空 |
| Maintain_WarehouseName | 维修仓库名称 | varchar(50) | 可空 |
| Maintain_Num | 维修数量 | int | 可空 |
| Maintain_Time | 维修时间 | varchar(50) | 可空 |

Ord_RecordTable: 军械巡查记录表, 用来记录巡查人随机巡查的信息与记录, 此表按照设计是有巡查人员进行填写, 在毕业设计要求当中, 巡查人员不定期进行军械维修保养或者入库等事务进行巡查, 并填写巡查记录以便部队各级领导进行查看。

表 3.4 Ord_RecordTable

| 字段名称 | 含义 | 类型 | 约束 |
|----------------|---------|---------------|----|
| Record_ID | 巡查记录 ID | int | 主键 |
| Record_UserID | 巡查人 ID | nvarchar(200) | 可空 |
| Record_Time | 巡查记录时间 | nvarchar(50) | 可空 |
| Record_Remarks | 巡查记录内容 | nvarchar(MAX) | 可空 |

Ord_ReportTable: 军械入库表, 用来记录所有的军械入库的详细信息。可以说此表是所有表中最为关键的一张表, 包含有入库军械的所有信息, 报废、维修、借用都是基于此表进行数据的修改, 在军械信息表当中添加完成之后, 进行入库操作会将填写入库的详细信息, 包括入库数量、入库仓库、入库批次等等, 上次维修周期默认为入库的当前时间。

表 3.5 Ord_ReportTable

| 字段名称 | 含义 | 类型 | 约束 |
|--------------------|---------|---------------|----|
| Report_ID | 入库 ID | int | 主键 |
| Report_ArmyID | 入库军械 ID | nvarchar(200) | 可空 |
| Report_WarehouseID | 入库仓库 ID | nvarchar(200) | 可空 |
| Report_Batch | 入库批次 | nvarchar(50) | 可空 |

| | | | |
|-----------------------|--------|----------------|----|
| Report_Num | 入库数量 | int | 可空 |
| Report_UserID | 操作员 ID | nvarchar(200) | 可空 |
| Report_Status | 入库状态 | int | 可空 |
| Report_CreatTime | 入库时间 | nvarchar(200) | 可空 |
| Report_Remarks | 入库标记 | nvarchar(1000) | 可空 |
| Report_InventoryNum | 可用数量 | int | 可空 |
| Report_BorrowNum | 借出数量 | int | 可空 |
| Report_ScrapNum | 报废数量 | int | 可空 |
| Report_TransportName | 运送人员姓名 | nvarchar(200) | 可空 |
| Report_TransportPhone | 运送联系方式 | nvarchar(50) | 可空 |
| Report_MaintainTime | 军械维修时间 | nvarchar(50) | 可空 |
| Report_MaintainStatus | 军械维修状态 | int | 可空 |

Ord_RoleTable: 军械角色表, 用来区分用户所属角色, 分别赋予不同权限。此表在页面当中按照系统需求只进行角色的显示, 而不进行角色的添加功能, 按照角色来对各个用户的权限进行区分, 这就防止出现用户权限越界操作不当而导致军械信息混乱的情形。

表 3.6 Ord_RoleTable

| 字段名称 | 含义 | 类型 | 约束 |
|-----------------|--------|---------------|----|
| Role_ID | 角色 ID | nvarchar(200) | 主键 |
| Role_Name | 角色名称 | nvarchar(50) | 可空 |
| Role_CreateTime | 角色创建时间 | nvarchar(200) | 可空 |
| Role_Status | 角色状态 | int | 可空 |

Ord_ScrapTable: 军械报废表, 用来记录所有的军械报废信息。报废信息的修改会对军械入库表中可借数量进行修改, 保持数据的一致性。

表 3.7 Ord_ScrapTable

| 字段名称 | 含义 | 类型 | 约束 |
|-------------------|---------|----------------|----|
| Scrap_ID | 报废 ID | int | 主键 |
| Scrap_ArmyID | 报废军械 ID | nvarchar(200) | 可空 |
| Scrap_WarehouseID | 报废仓库 ID | nvarchar(200) | 可空 |
| Scrap_Batch | 报废军械批次 | nvarchar(50) | 可空 |
| Scrap_Num | 报废数量 | int | 可空 |
| Scrap_UserID | 报废用户 ID | nvarchar(200) | 可空 |
| Scrap_Time | 报废时间 | nvarchar(200) | 可空 |
| Scrap_Remarks | 报废标记 | nvarchar(1000) | 可空 |

Ord_UserTable: 用户信息表, 用来记录所有的用户信息。按照角色表分别进行权限的赋予, 在创建用户的时候, 用户密码的设定进行了限制, 必须由数字字母六位数组成, 防止在创建之后有恶意用户尝试登录系统, 保证安全性。并且用户还设有了启用以及停用操作, 如果某用户操作不当可以对其进行禁用。

表 3.8 Ord_UserTable

| 字段名称 | 含义 | 类型 | 约束 |
|-----------------|--------|---------------|-----|
| User_ID | 用户 ID | nvarchar(200) | 主键 |
| User_PassWord | 用户密码 | nvarchar(50) | 可空 |
| User_Name | 用户姓名 | nvarchar(50) | 可空 |
| Dept_ID | 部门 ID | nvarchar(200) | 可空 |
| Role_ID | 角色 ID | nvarchar(200) | 可空 |
| User_Status | 用户状态 | int | 不可空 |
| User_CreateTime | 用户创建时间 | nvarchar(200) | 可空 |

Ord_WarehouseRemarks: 军械仓库表, 用来记录所有的仓库信息。军械的仓库表, 仓库应该有系统管理员创建, 默认一个仓库只包含有一个仓库管理员账号。

表 3.7 Ord_WarehouseRemarks

| 字段名称 | 含义 | 类型 | 约束 |
|----------------------|---------|----------------|----|
| Warehouse_ID | 仓库 ID | nvarchar(200) | 主键 |
| Warehouse_Name | 仓库名称 | nvarchar(50) | 可空 |
| Warehouse_UserID | 仓库用户 ID | nvarchar(200) | 可空 |
| Warehouse_Location | 仓库地址 | nvarchar(200) | 可空 |
| Warehouse_Area | 仓库面积 | int | 可空 |
| Warehouse_CreateTime | 仓库创建时间 | nvarchar(50) | 可空 |
| Warehouse_Remarks | 仓库标记 | nvarchar(1000) | 可空 |

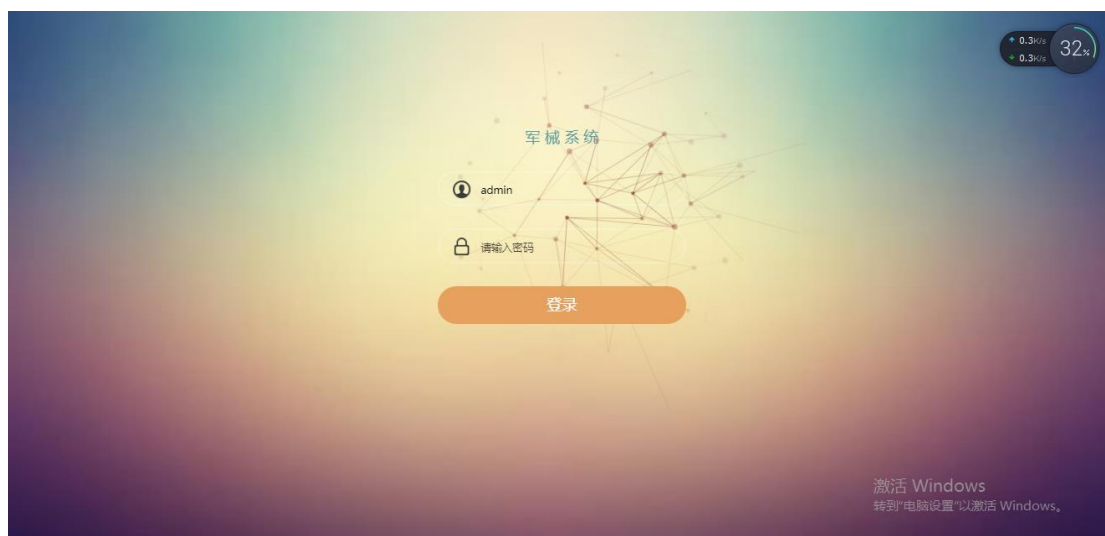
4 详细设计及实现

4.1 登录

4.1.1 登录认证

(1) 界面

图 4.1 登录界面图示



(2) 功能

运行军械存储保障管理系统之后，进入登录界面，按照提示输入用户名和用户密码登陆本系统，在系统后台根据用户 ID 进行区分用户，用以屏蔽不同的按键以及功能。

(3) 编程思路

本军械装备存储保障系统按照用户名、用户密码、用户状态进行登录验证。如果当前用户处于停用状态，那么就会提示用户已停用；此时进行用户名的验证，搜索数据库中是否含有本用户，如果不存在此用户会提示用户不存在，若是密码输入错误则会提示密码错误。

(4) 主要代码

```
{  
    dt = indexDal.GetLoginLoggerData(username, password);  
    IList<UserModel> userlist = new List<UserModel>();  
    userlist =
```

```
UserController.ModelConvertHelper<UserModel>.ConvertToModel(dt);
    bool num = userlist.Count > 0;
    if (userlist.Count > 0)
    {
        Role_Id = userlist[0].Role_ID;
        User_Id = userlist[0].User_ID;
        User_Name = userlist[0].User_Name;
        User_Status = userlist[0].User_Status;
    }
}
```

4.2 首页信息展示

4.2.1 月度军械装备存储保障记录

(1) 界面

图 4.2 军械统计折线图



(2) 功能

如图所示，显示本月度的军械入库数量以及军械总的统计数量；显示月度军械借出数量以及总的借出数量；显示月度军械报废数量以及总的报废数量；显示正常用户数量以及总的用户数量。

(3) 编程思路

登录之后，调用 ajax，执行逻辑查询查询，首先到军械入库总表中查询出所有数据，生成总入库数量，然后使用 linq 进行条件查询，筛选出本月的入库信息，进行求和，生成月度入库数量；再者查询军械借出信息，首先查询出所有借出信息，进行汇总，获取总的借出数量，再使用 linq 进行条件查询生成月度借出数量；其次查询军械报废信息，首先获取所有报废信息，进行汇总获取总的报废数量，然后进行 linq 的条件查询获取本月度的报废总数；最后针对于用户，按照用户状态查询出总的用户以及正常用户状态。将这八个数据以 json 包返回到前台页面

进行解析，按照 ID 获取数据显示到页面当中。

(4) 主要代码

```
public ActionResult GetAllData()
{
    //获取入库的总数据 用linq再次查询总数
    IList<ReportModel> userlist1 = new List<ReportModel>();
    userlist1 =
UserController.ModelConvertHelper<ReportModel>.ConvertToModel(dt1);
    zruku = userlist1.Sum(a => a.Report_Num);

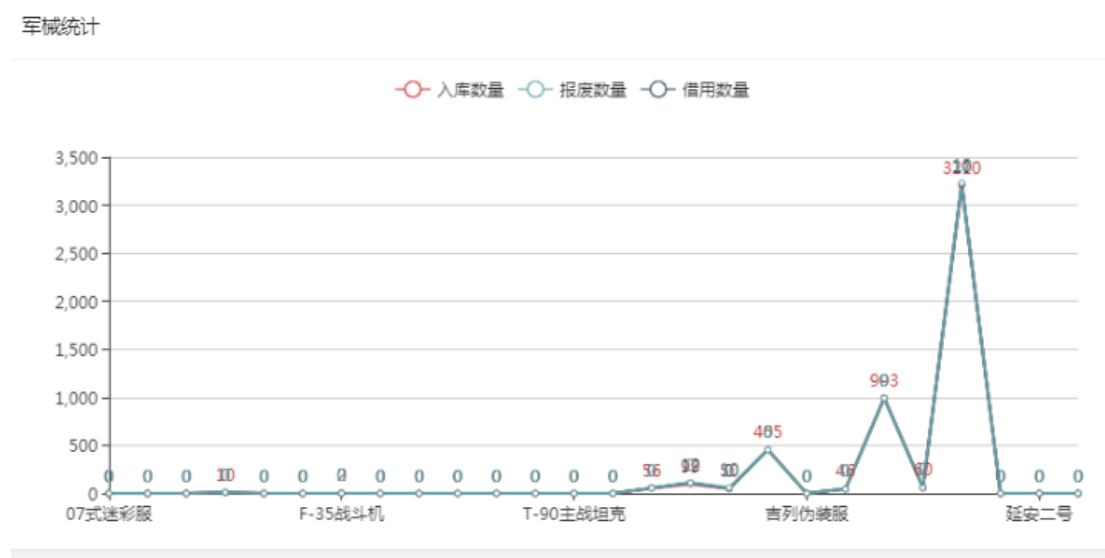
    //获取借出总数据 用linq再次查询总数以及月度数
UserController.ModelConvertHelper<BorrowModel>.ConvertToModel(dt2);
    zchuku = userlist2.Sum(a => a.Borrow_Num);

    //获取报废总数 用linq再次查询总数以及月度数
    IList<ScrapModel> userlist3 = new List<ScrapModel>();
    userlist3 =
UserController.ModelConvertHelper<ScrapModel>.ConvertToModel(dt3);
    zbaofei = userlist3.Sum(a => a.Scrap_Num);
}
```

4.2.2 军械数据统计图表

(1) 界面

图 4.3 军械统计折线图



(2) 功能

军械信息总的图表主要是为了展示各个军械总的出入库以及报废情况，给部队人员一种更为清楚详细直观的军械信息使用展示。包括有军械名称、军械在库数量、军械借出数量、军械报废数量。

(3) 编程思路

使用 echart 工具生成图表，方便快捷，导入数据即可使用，后台先分别获取军械名称，军械入库数量、军械借出数量、以及军械的报废数量，使用 list 列表接受各个数据，然后以 json 包的形式打回前台图表页面，传递数据，生成定制图表。

(4) 主要代码

```
public ActionResult HomeChart(string name)
{
    dt = armyDal.GetHomeChartData();
    List<string> data = new List<string>();
    List<string> seriesRuku = new List<string>();
    List<string> seriesJEku = new List<string>();
    List<string> seriesBF = new List<string>();
}
```

4.3 军械管理界面

4.3.1 军械信息页面

(1) 界面

图 4.4 军械信息记录图

军械名称

添加

| <input type="checkbox"/> | 军械名称 | 维护周期（月） | 创建时间 | 操作 |
|--------------------------|----------|---------|------------|---------------------------------------|
| <input type="checkbox"/> | 哨兵R-1 | 7 | 2019-05-13 | 编辑 删除 |
| <input type="checkbox"/> | 东风猛士 | 6 | 2019-05-13 | 编辑 删除 |
| <input type="checkbox"/> | 歼-31 | 10 | 2019-05-13 | 编辑 删除 |
| <input type="checkbox"/> | 山猫侦察车 | 5 | 2019-05-13 | 编辑 删除 |
| <input type="checkbox"/> | F-35战斗机 | 6 | 2019-05-13 | 编辑 删除 |
| <input type="checkbox"/> | 巴雷特M82A1 | 1 | 2019-05-13 | 编辑 删除 |
| <input type="checkbox"/> | 96A式主战坦克 | 4 | 2019-05-13 | 编辑 删除 |
| <input type="checkbox"/> | 虎牙救生刀 | 6 | 2019-05-13 | 编辑 删除 |
| <input type="checkbox"/> | 07式迷彩服 | 12 | 2019-05-13 | 编辑 删除 |

图 4.5 军械信息添加图

（2）功能

军械信息界面用于展示添加的军械信息（注意不是入库信息），军械首先要有军械信息才能有入库信息。点击添加按钮，输入军械名称以及军械的维护周期，这样军械信息就会添加一条记录，显示军械名称、维护周期、创建时间。只有未进行入库的军械信息可以进行编辑和删除。

（3）编程思路

首先查询军械信息表，获取所有的军械信息的所有数据，json 包返回到前台页面，前台页面进行解析，显示出军械名称、军械维护周期、创建时间以及操作等等。并且对于军械可以按照关键词进行搜索，获取相符的军械信息。只有军械其他所有信息为空的时候才能进行军械信息的编辑和删除。点击添加按钮的时候，弹出添加按钮的信息，进行军械信息的添加，相应的点击编辑会出现军械信息的编辑页面，进行数据记录的修改编辑。

（4）主要代码

```
public ActionResult GetArmyData(string Army_Name)
{
    List<ArmyModel> userlist = new List<ArmyModel>();
    userlist =
UserController.ModelConvertHelper<ArmyModel>.ConvertToModel(dt).ToList();

    return Json(new { code = 0, count = userlist.Count, data = userlist },
JsonRequestBehavior.AllowGet);
}
```

4.3.2 军械数据页面

(1) 界面

表 4.6 军械数据图

军械管理系统

军械管理

军械信息

军械数据

入库管理

借用管理

维护管理

报废管理

巡查管理

基础管理

admin

军械名称 请输入 所属仓库 请选择 日期范围

| 军械名称 | 军械批次 | 仓库名称 | 军械数量 | 在库数量 | 借出数量 | 报废数量 | 日期 | 军械状态 |
|-----------|------------|-----------|------|------|------|------|------------|------|
| 响兵R-1 | ABCD-01 | 仓库CK-2001 | 100 | 78 | 10 | 12 | 2019-05-13 | 正常 |
| 东风猛士 | ABCD-03 | 仓库CK-2012 | 99 | 87 | 12 | 0 | 2019-05-13 | 正常 |
| 歼-31 | EDCF-2365 | 仓库CK-2013 | 12 | 11 | 1 | 0 | 2019-05-13 | 正常 |
| 响兵R-1 | xtx | 仓库CK-2013 | 1555 | 1555 | 0 | 0 | 2019-05-13 | 正常 |
| 响兵R-1 | xtx | 仓库CK-2013 | 1555 | 1555 | 0 | 0 | 2019-05-13 | 正常 |
| F-35战斗机 | BHJL-1222 | 仓库CK-1004 | 2 | 2 | 0 | 0 | 2019-05-13 | 正常 |
| 巴雷特M82A1 | SGHJ-4565 | 仓库CK-2013 | 56 | 56 | 0 | 0 | 2019-05-13 | 正常 |
| 96A式主战... | DSFG-2852 | 仓库CK-2012 | 10 | 10 | 0 | 0 | 2019-05-13 | 正常 |
| 虎牙教生刀 | FDSF-46545 | 仓库CK-2001 | 455 | 455 | 0 | 0 | 2019-05-13 | 正常 |
| 山猫侦察车 | GJHG-4564 | 仓库CK-2012 | 60 | 50 | 10 | 0 | 2019-05-13 | 正常 |
| 沙漠之鹰 | GJHG-4555 | 仓库CK-1004 | 993 | 993 | 0 | 0 | 2019-05-13 | 正常 |
| 悍马 | GHGJ-5454 | 仓库CK-2012 | 50 | 40 | 0 | 10 | 2019-05-13 | 正常 |
| 歼-31 | FHGF-8787 | 仓库CK-1004 | 34 | 34 | 0 | 0 | 2019-05-15 | 正常 |

(2) 功能

本界面是军械信入库之后的总界面，可以按照军械名称、所属仓库以及日期范围进行军械入库信息的条件查询。本界面显示有军械名称、军械批次、仓库名称、军械数量、在库数量、借出数量、报废数量、日期以及军械目前的状态。

(3) 编程思路

在底层进行联合查询，附带军械名称、所属仓库以及日期，在总的入库信息表中进行查询，查出军械名称、军械批次、仓库名称、军械数量、在库数量、借出数量、报废数量、日期以及军械状态，查询完毕之后，以 json 包返回前天进行解析，在 layui 的 table 中按照 ID 进行数据的填充。

(4) 主要代码

```
public ActionResult GetWarehouseData(string Warehouse_Name)
{
    dt = armyDal.GetWarehouseData(Warehouse_Name);
    IList<WarehouseModel> userlist = new List<WarehouseModel>();
    userlist =
UserController.ModelConvertHelper<WarehouseModel>.ConvertToModel(dt);
}
```

4.4 入库管理

4.4.1 入库信息

(1) 界面

图 4.7 入库信息图

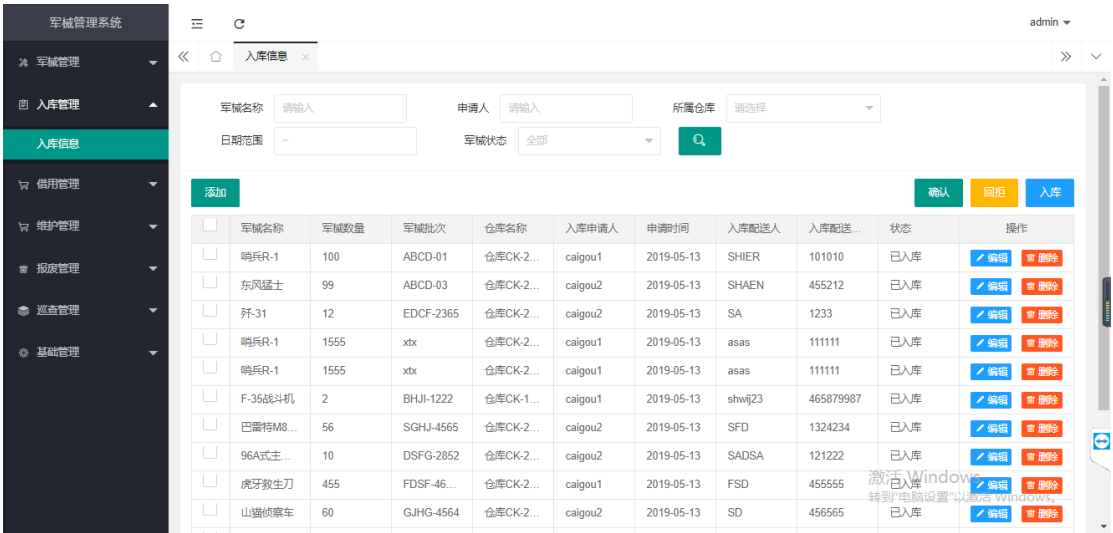


图 4.8 采购入库图

采购入库

军械名称

请选择

入库仓库

请选择

军械批次

请输入批次

军械数量

请输入数量

采购员

请选择

配送人姓名

请输入配送人姓名

联系方式

请输入联系方式

确定

取消

（2）功能

实现军械入库信息的显示，能够按照军械名称、申请人、所属仓库、日期范围、军械状态进行条件查询。在记录显示之中，含有军械名称、军械数量、军械批次、仓库名称、入库申请人、入库时间、入库配送人、入库配送人联系方式、入库状态。点击添加按钮可以进行军械数据信息的添加，包括军械名称、入库仓库、军械批次、军械数量、采购员、配送人以及配送人联系方式。只有已提交的军械数据可以进行军械数据的修改，对于已确认、已回拒和已入库的军械信息则不能够进行军械信息的修改。

（3）编程思路

对于军械入库信息的显示，从入库信息表中查询到数据之后，以 json 包形式返回前台界面进行解析，对于按照军械名称、申请人、所属仓库、日期范围、军械状态进行条件查询是与军械信息查询同步的，只是开始时查询条件为空。对于军械入库信息的添加，接收到军械数据之后，进行军械入库表记录的添加，然后刷新页面显示到界面当中。

（4）主要代码

```
data: {
    Report_ArmyID: field.Report_ArmyID,
    Report_WarehouseID: field.Report_WarehouseID,
    Report_Batch:field.Report_Batch,
    Report_Num: field.Report_Num,
    Report_UserID:field.Report_UserID ,
    Report_TransportName:field.Report_TransportName ,
    Report_TransportPhone:field.Report_TransportPhone
},
```

4.5 借用管理

4.5.1 借用信息

（1）界面

图 4.9 借用信息图

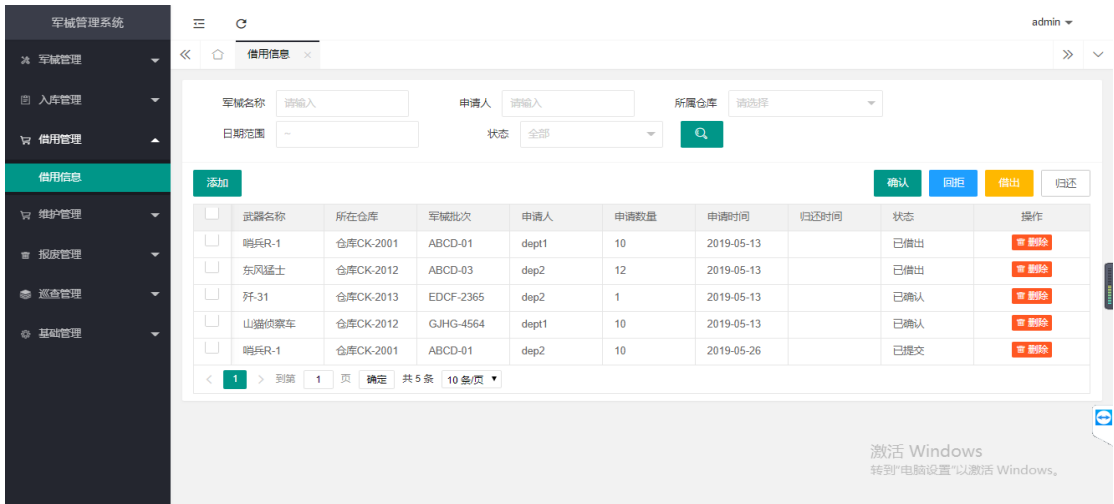


图 4.10 借用信息添加图

采购入库

军械名称

请选择

借用仓库

请选择

军械批次

请选择

可借数量

借用数量

请输入数量

借用人

请选择

提取人姓名

请输入配送人姓名

联系方式

请输入联系方式

确定

取消

(2) 功能

本界面主要是实现军械信息的借用管理，即出库管理。可以按照军械名称、申请人、所属仓库、日期范围、状态进行军械出库信息的查询。对于军械借出记录显示有武器名称、所在仓库、军械批次、申请人、申请数量、申请时间、返还时间等。点击添加按钮进行军械借出信息的添加，只有在库的军械才可以进行军械的借出操作。选择军械名称、借用仓库、借用批次、显示可借数量、借用数量、

借用人、运送人员以及运送人员联系方式，点击确定进行借用信息添加。

(3) 编程思路。

对于军械借出信息的显示，以军械名称、申请人、所属仓库、日期范围、状态进行条件查询，首次显示时条件默认为空将所有军械借出信息查询出来，以 json 包返回前台进行解析。对于军械信息的借出，含有多级联动，选择军械名称之后会按照军械名称查询那些仓库含有此类军械，选择仓库后会查询含有那些军械批次，自动显示出可借出数量，然后进行配送人信息的填写。借用信息提交之后，可以进行借用记录的修改，显示借用信息的状态。并且提交之后要受到仓库管理员的审核，如果不符合要求规定，那么仓库管理员就会拒绝本次的军械借出申请。部门操作员选择一条军械借出的数据，可以进行军械归还。

(4) 主要代码

```
{
    List<BorrowModel> userlist = new List<BorrowModel>();
    userlist =
UserController.ModelConvertHelper<BorrowModel>.ConvertToModel(dt).ToList();
}

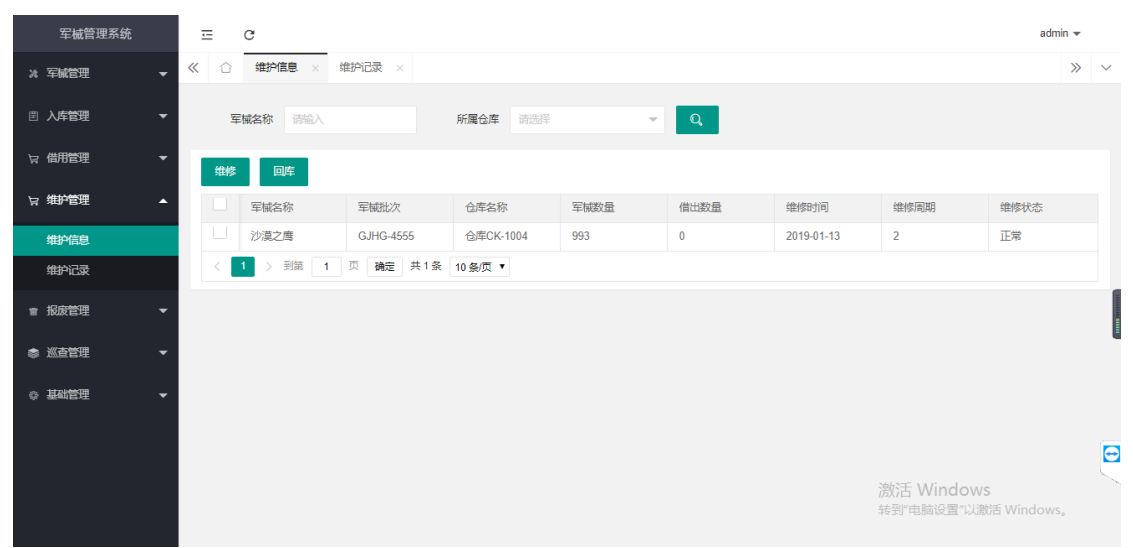
public ActionResult AddBorrow(BorrowModel borrowModel)
{
    borrowModel.Borrow_Status = 0;
    borrowModel.Borrow_Time = DateTime.Now.ToString("yyyy-MM-
dd");
}
```

4.6 维护管理

4.6.1 维护信息

(1) 界面

图 4.11 维护信息图



(2) 功能

本界面主要是实现了维修信息的显示，可以按照军械名称以及所属仓库进行军械维修记录的查询，在最初的军械信息添加的时候，添加了此军械的维护周期（以月为单位）。维修计划表显示有军械名称、军械批次、仓库名称、军械数量、借出数量、维修时间、维修周期以及维修状态。选中一条待维修的记录之后，点击维修按钮，此时会修改维修状态，将正常状态换为维修中，当再次点击维修中的状态的记录时，会默认为此时维修完成需要返回仓库。

(3) 编程思路

首次加载此页面的时候，默认此时的军械名称、所属仓库等条件为空，查询出所有超过需要维修周期的军械信息，但是默认情况下，如果军械此时有借出记录没有入库的话，也是不能显示在维修计划表当中的。点击维修按钮时，会选中当前记录，进行当前军械记录维修状态的修改，如果已经在维修中那么会给出相应的提示。点击回库按钮的时候，会进行维修时间的修改，将上次维修时间修改为当前时间，并生成维修记录。后台根据维修周期在入库表中进行条件查询。

(4) 主要代码

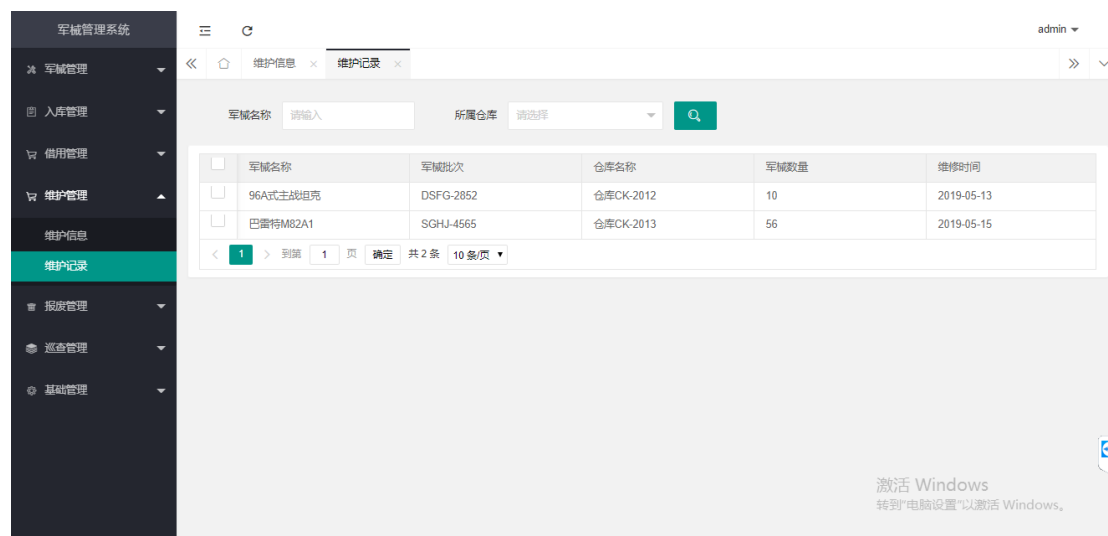
```
{
    List<ReportModel> userlist = new List<ReportModel>();
    userlist =
UserController.ModelConvertHelper<ReportModel>.ConvertToModel(dt).ToList();
    userlist = userlist.Where(a =>
```

```
Convert.ToDateTime(a.Report_MaintainTime).AddMonths(a.Report_RepairCycle) <
DateTime.Now).ToList();
    }
```

4.6.2 维护记录

(1) 界面

图 4.12 维护记录图



(2) 功能

实现军械维修记录的显示，包含军械名称、军械批次、仓库名称，军械数量以及维修时间。显示所有的维修记录，可以按照军械名称或者所属仓库进行军械维修记录的条件查询。

(3) 编程思路

主要是从军械维修记录表中查询数据然后显示到界面当中，对于维护记录，在维护信息中当军械武器维修完成之后点击回库按钮，进行武器会回库，此时还会生成维修记录。后台通过查询维修记录表，然后以 json 包的形式返回前台页面，按照械名称、军械批次、仓库名称，军械数量以及维修时间进行表格的填充显示。

(4) 主要代码

```
{
```

```

        List<MaintainModel> userlist = new List<MaintainModel>();
        userlist =
UserController.ModelConvertHelper<MaintainModel>.ConvertToModel(dt).ToList()
    }

```

4.7 报废管理

4.7.1 报废信息显示

(1) 界面

图 4.13 报废信息图



(2) 功能

显示近期的报废记录，包含军械名称、仓库名称、军械批次、报废数量以及报废时间。可以按照军械名称、所属仓库以及日期范围进行军械报废信息的条件查询。

(3) 编程思路

军械报废信息添加以后，首次加载时可以按照军械名称、所属仓库以及日期范围默认为空进行条件查询，查询之后以 json 包形式返回前台表格显示。

(4) 主要代码

```

{
    dt = scrapDal.GetScrapData(UserID , Scrap_ArmyName,
Scrap_WarehouseID, Scrap_Time);
    List<ScrapModel> userlist = new List<ScrapModel>();

```

}

4.7.2 报废信息添加

(1) 界面

图 4.14 报废信息添加图

(2) 功能

此弹窗主要用于军械报废记录的添加,包含军械名称、报废仓库、军械批次、在库数量、报废数量。点击确定按钮之后回生成报废记录。

(3) 编程思路

此弹窗是属于多级联动的,首先需要在军械信息中有此军械的入库信息,并且选择入库军械之后,会查询那些仓库含有此军械的信息,其次会查看此仓库中拥有哪些军械批次,再者会查出在库数量,编辑报废数量之后回提交报废记录,报废表中生成相应的记录,并且会修改入库表中的军械信息可借出数量。

(4) 主要代码

```
{
    scrapModel.Scrap_Time = DateTime.Now.ToString("yyyy-MM-dd");

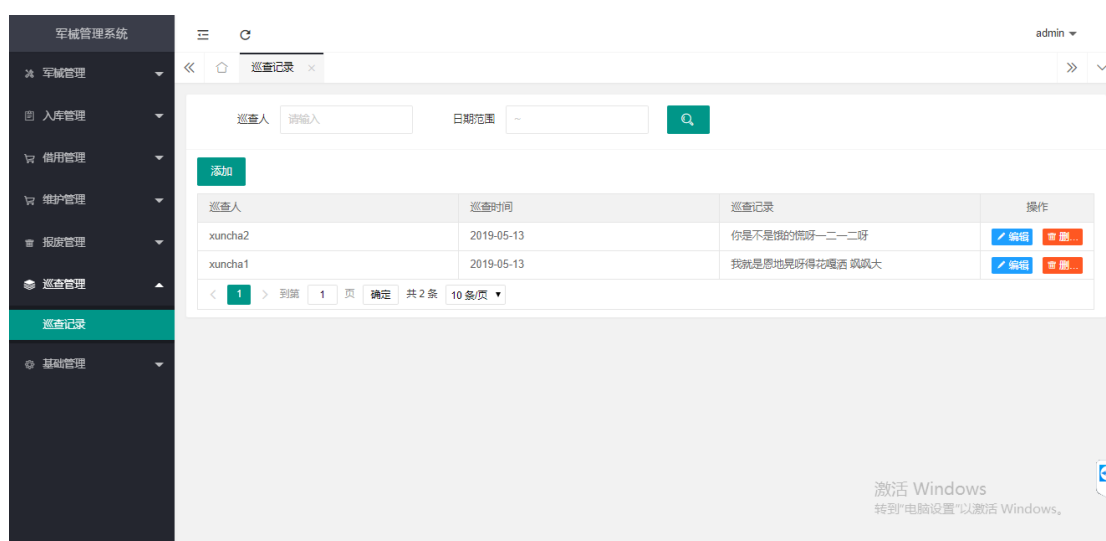
    ScrapDal armyDal = new ScrapDal();
    bool dt = armyDal.AddScrap(scrapModel);
    return Json(dt);
}
```

4.8 巡查管理

4.8.1 巡查记录显示

(1) 界面

图 4.15 巡查信息图



(2) 功能

此界面是显示巡查记录的，可以按照巡查人以及日期范围来进行巡查记录的分类。包含有巡查人、巡查时间以及巡查内容。

(3) 编程思路

按照用户的角色 ID 来进行区分，只有巡查人才能进行巡查记录的添加、编辑和删除巡查记录添加完成后，部队领导人员可以进行巡查记录的浏览和查看。

当巡查人员进行巡查记录的编辑时修改记录表中的数据，然后会进行异步提交。

(4) 主要代码

```
public ActionResult GetRecordData( string Record_UserName,string Record_Time)
{
    dt = recordDal.GetRecordData("", Record_UserName, Record_Time);
    List<RecordModel> userlist = new List<RecordModel>();
    userlist =
UserController.ModelConvertHelper<RecordModel>.ConvertToModel(dt).ToList();
}
```

4.8.2 巡查记录添加

(1) 界面

图 4.16 巡查记录新增图



巡查记录新增

巡查人 请选择

维护记录 请输入内容

确定 取消

(2) 功能

此页面是实现巡查记录新增的功能，巡查人员是系统管理员在后台添加的，选择巡查人之后，输入维护记录，点击确定按钮就会生成一条巡查记录。

(3) 编程思路

巡查人员登录之后，进入巡查管理界面，点击添加记录的按钮，会弹出提示框，要求选择巡查人以及添加维护记录，编辑完成后，应当修改巡查记录表，进行巡查记录的添加。后台执行查询函数，然后以 json 包形式返回前台，填充表

格。

(4) 主要代码

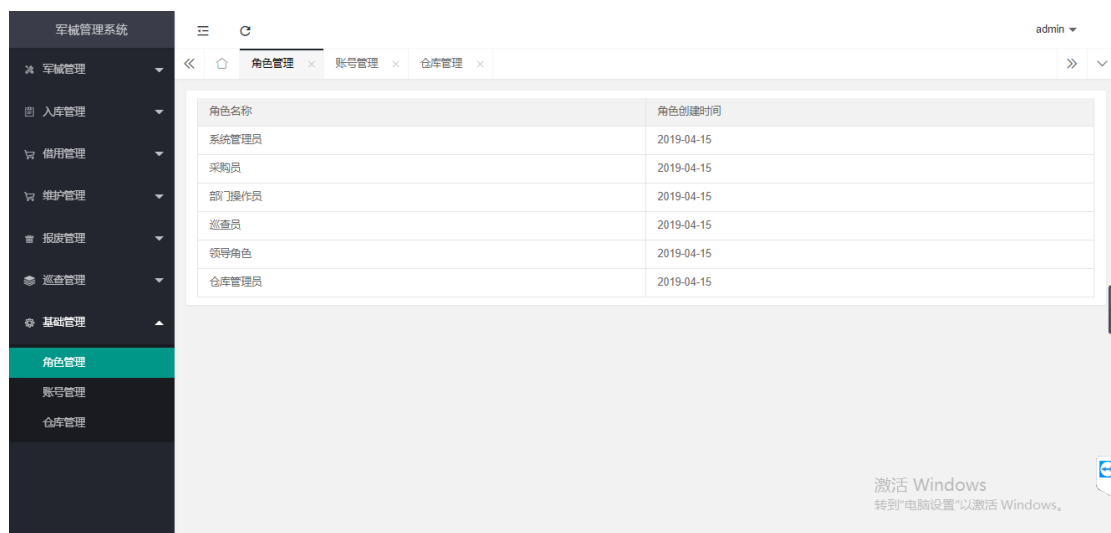
```
{
    if (type == "2")
    {
        DataTable dt = new DataTable();
        RecordDal recordDal = new RecordDal();
        dt = recordDal.GetRecordData(ID, "", "");
        List<RecordModel> userlist = new List<RecordModel>();
        ViewBag.UserID = userlist[0].Record_UserID;
        ViewBag.Remarks = userlist[0].Record_Remarks;
    }
}
```

4.9 基础管理

4.9.1 角色管理

(1) 界面

图 4.17 角色信息图



| 角色名称 | 角色创建时间 |
|-------|------------|
| 系统管理员 | 2019-04-15 |
| 采购员 | 2019-04-15 |
| 门卫操作员 | 2019-04-15 |
| 巡查员 | 2019-04-15 |
| 领导角色 | 2019-04-15 |
| 仓库管理员 | 2019-04-15 |

(2) 功能

角色管理界面是显示当前系统中存在的各种部队人员角色, 包含系统管理员

(admin)、采购员、部门操作员、巡查人员、仓库管理员以及领导人员。并且本系统是按照此用户角色来进行军械存储保障管理系统的功能以及按钮的屏蔽,防止出现越级行为。

(3) 编程思路

角色管理界面当中所有的角色都是在系统后台的数据库当中默认添加好了的,在用户添加过程中,防止出现角色不明显的现象,包含角色名称以及角色创建时间。在后台查询角色表,以 json 包形式返回前台界面,并填充表格。

(4) 主要代码

```
{
    dt = userdal.GetRole();
    List<RoleModel> userlist = new List<RoleModel>();
    userlist =
ModelConvertHelper<RoleModel>.ConvertToModel(dt).ToList();
}
```

4.9.2 账号管理

(1) 界面

图 4.18 账号信息图

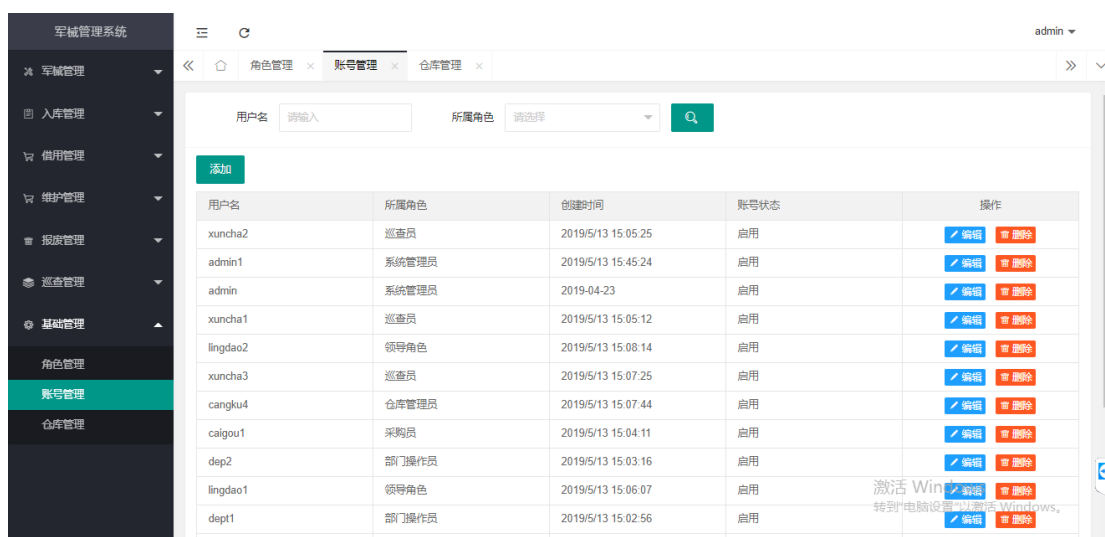


图 4.19 用户信息添加图

（2）功能

账号管理界面，实现各个用户信息的显示，包括用户名、所属角色、创建时间以及用户状态。可以按照用户名以及所属角色进行条件查询。在添加用户时，输入用户名，输入用户密码，选择所属角色，以及是否启用该用户，如果该用户未启用则在登录本系统的时候会提示用户处于停用状态。

（3）编程思路

首先对于用户信息展示来说，需要获取用户的所有信息，包括用户名、所属角色、创建时间以及用户状态。后台获取数据之后，返回 json 包解析然后填充所有表格。对于用户角色添加，用户名没有什么限制，但是对于用户密码，在前台设置了一个正则表达式来限定用户密码的输入必须符合字母数字混合输入，否则会给出提示。选择角色以及是否启用之后，点击确定按钮会进行数据库中用户表记录的添加。点击编辑按钮会进行账号信息的修改和编辑，包括用户名和用户密码。

（4）主要代码

```
{
```

```

dt = userdal.GetUser(ID, "", "");
IList<UserModel> userlist = new List<UserModel>();
ViewBag.User_ID = userlist[0].User_ID;
ViewBag.User_Name = userlist[0].User_Name;
ViewBag.User_PassWord = userlist[0].User_PassWord;
ViewBag.User_RoleID= userlist[0].Role_ID;
ViewBag.User_Status= userlist[0].User_Status;
    }

```

4.9.3 仓库管理

(1) 界面

图 4.20 仓库信息图

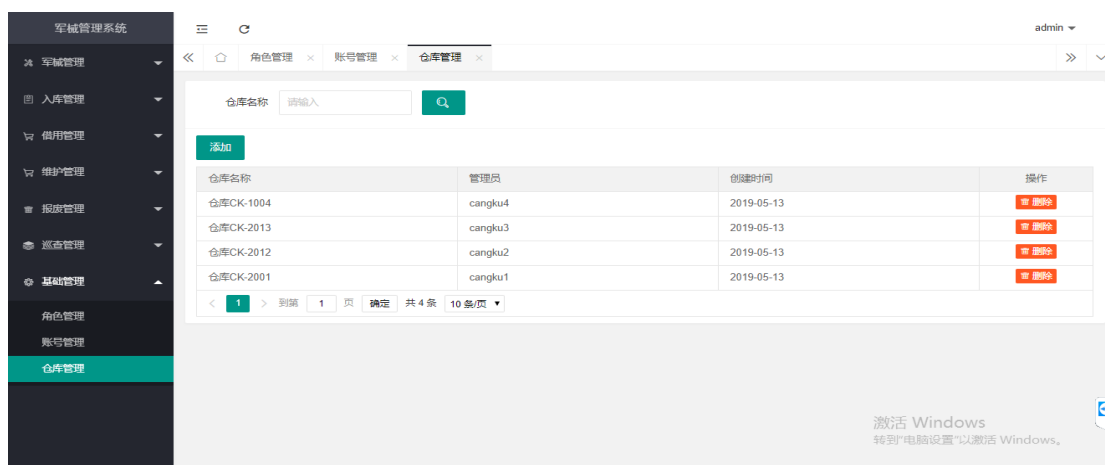


图 4.21 仓库信息添加图

The screenshot shows the 'Add Warehouse' (仓库新增) dialog box. It contains two input fields: 'Warehouse Name' (仓库名称) with a placeholder 'Please enter warehouse name' (请输入仓库名称), and 'Manager' (管理员) with a dropdown menu showing 'Please select' (请选择). At the bottom right, there are two buttons: 'Confirm' (确定) and 'Cancel' (取消).

（2）功能

实现仓库信息的显示以及添加。仓库信息的显示包含仓库名称、管理员、仓库创建时间。点击添加按钮可以进行仓库的添加，一个仓库默认只属于一个管理员。点击添加弹出仓库添加的窗体，输入仓库名称，选择仓库管理员，点击确定按钮，在仓库表中进行新增仓库记录的添加。

（3）编程思路

对于仓库信息页面，后台查询数据库仓库标的信息，以 json 包形式将数据返还，解析之后填充到表格当中。对于添加按钮，类似于账号管理的添加，输入仓库名称，然后选择管理员，点击确定之后会在系统的仓库表中进行仓库新增一条记录。

（4）主要代码

```
{
    if (type == "2")
    {
        DataTable dt = new DataTable();
        RecordDal recordDal = new RecordDal();
        dt = recordDal.WarhouseDatas(ID);
        List<WarehouseModel> userlist = newList<WarehouseModel>();
        ViewBag.Warehouse_UserID = userlist[0].Warehouse_UserID;
        ViewBag.Warehouse_Name = userlist[0].Warehouse_Name;
    }
}
```

5 系统测试

5.1 测试目的

在实际应用当中，我们发现如果跳过测试步骤直接下发新版本到服务器上，会不可避免的出现大量 bug，导致用户的体验以及用户使用效率大大降低，这样难以保证软件以及系统的质量，很容易造成客户以及用户的流失。为了保证本军械存储保障管理系统的可靠性与安全性，必须进行可靠有序的系统测试，借以保证本系统的效率以及安全。

5.2 测试用例

表 5.1 登录认证测试表

| 测试模块：登录认证 | |
|-----------|---|
| 测试内容 | <ol style="list-style-type: none"> 1. 正常用户是否能正常登陆 2. 不存在用户错误是否给出提示 3. 非正常状态用户是否给出提示 4. 密码错误是否给出提示 |
| 测试输入 | <ol style="list-style-type: none"> 1. 正常用户 2. 不存在用户 3. 非正常状态用户 4. 密码错误 |
| 预期结果 | <ol style="list-style-type: none"> 1. 正常用户正常登陆 2. 问题用户分别给出提示信息 |
| 实际结果 | <ol style="list-style-type: none"> 1. 正常用户正常登陆 2. 问题用户分别给出提示信息 |

表 5.2 首页信息展示测试表

| 测试模块：首页信息展示 | |
|-------------|--|
| 测试内容 | 1. 月度军械装备存储保障数据是否显示 2. 军械数据统计图表是否正常显示 |
| 测试输入 | |
| 预期结果 | 1.月度军械装备存储保障数据显示正常 2.军械数据统计图正常显示 |
| 实际结果 | 1. 月度军械装备存储保障数据显示正常 2. 军械数据统计图正常显示 |

表 5.3 军械管理测试表

| 测试模块：军械管理 | |
|-----------|--------------------------------------|
| 测试内容 | 1. 军械信息查询展示是否正常 2. 军械信息是否正常添加以及显示 |
| 测试输入 | 1.添加军械数据以及军械维护周期 |
| 预期结果 | 1. 军械信息能够正常查询显示 2. 军械信息能够正常添加并且显示 |
| 实际结果 | 1. 军械信息能够正常查询显示 2. 军械信息能够正常添加并且显示 |

表 5.4 入库管理测试表

| 测试模块：入库管理 | |
|-----------|---|
| 测试内容 | 1. 军械入库信息能否正常查询显示 2. 军械入库信息是否能正常添加、确认、回拒以及入库 |

| | |
|------|--|
| 测试输入 | 1.添加军械入库数据 |
| 预期结果 | 1. 军械入库信息能正常查询显示 2. 军械入库信息是否能正常添加、确认、回拒以及入库 |
| 实际结果 | 1. 军械入库信息能正常查询显示 2. 军械入库信息是否能正常添加、确认、回拒以及入库 |

表 5.5 借出管理测试表

| 测试模块：借出管理 | |
|-----------|----------------------------------|
| 测试内容 | 1. 借用信息能否正常查询显示 2. 借用信息能否正常添加 |
| 测试输入 | 1. 添加军械借出记录 |
| 预期结果 | 1. 借用信息能够正常查询显示 2. 借用信息能够正常添加 |
| 实际结果 | 1. 借用信息能够正常查询显示 2. 借用信息能够正常添加 |

表 5.6 维护管理测试表

| 测试模块：维护管理 | |
|-----------|---|
| 测试内容 | 1. 测试维护信息能否正常显示 2. 需要维护军械能否正常维护 3. 维护记录能否正常添加 |
| 测试输入 | |
| 预期结果 | 1. 测试计划表能够正常显示 2. 需要维护军械能够正常维护 3. 维护记录能正常添加 |
| 实际结果 | 1. 测试计划表能够正常显示 |

| | |
|--|---------------------------------|
| | 2. 需要维护军械能够正常维护 3. 维护记录能正常添加 |
|--|---------------------------------|

表 5.7 报废管理测试表

| 测试模块：报废管理 | |
|-----------|--------------------------------|
| 测试内容 | 1. 报废信息能否正常显示 2. 报废信息能否正常添加 |
| 测试输入 | 1. 添加报废信息 |
| 预期结果 | 1. 报废信息能正常显示 2. 报废信息能正常添加 |
| 实际结果 | 1. 报废信息能正常显示 2. 报废信息能正常显示 |

表 5.8 巡查管理测试表

| 测试模块：巡查管理 | |
|-----------|--------------------------------|
| 测试内容 | 1. 巡查记录能否正产显示 2. 巡查记录能否正常添加 |
| 测试输入 | 1. 巡查记录的填写 |
| 预期结果 | 1. 巡查记录能正常显示 2. 巡查记录能正常添加 |
| 实际结果 | 1. 巡查记录能正常显示 2. 巡查记录能正常添加 |

表 5.9 基础管理测试表

| 测试模块：基础管理 | |
|-----------|---|
| 测试内容 | 1. 角色能否正常显示 2. 账号信息的显示与添加 3. 仓库信息的显示与添加 |

| | |
|------|---|
| 测试输入 | <ol style="list-style-type: none"> 1. 账号信息添加 2. 仓库信息添加 |
| 预期结果 | <ol style="list-style-type: none"> 1. 角色能正常显示 2. 账号信息能正常显示以及添加 3. 仓库信息能正常显示与添加 |
| 实际结果 | <ol style="list-style-type: none"> 1. 角色能正常显示 2. 账号信息能正常显示以及添加 3. 仓库信息能正常显示与添加 |

6 结论

本次毕业设计是针对部队需求结合科学实践而开发了一个军械存储保障管理系统。本系统有着军械信息添加、军械入库、军械出库、军械维护、军械巡查还有基础管理等一系列功能，能够满足部队各级部门的需求，及时准确的反馈军械出入库情况以及军械各种维修记录等等，防止出现军械丢失、军械无记录等严重情况的发生。

在设计初期，由于没有数据库设计的经验，对于数据库的设计让我很难下手，最初的数据库只包含了 ID、名称、时间此类，但随着设计的深入发现仅仅是简单的设计并不能满足所有需要，还会产生数据上的混乱，例如针对与同类军械的入库到不同仓库、同类军械入库到相同仓库借用时怎么区分这一情况，在查阅资料之后发现可以通过军械批次来进行区分。又如关于维修计划表，一开始的想法是创建一个实体表，到维修时间就插入进去，但是后来发现频繁的数据增删只会增加设计的复杂性，所以就进行了维护周期的设定，然后通过设定维修字段，只按照此条件以及借出数量进行查询操作生成虚拟维修计划表，大大降低了设计的复杂性。

通过本次毕业设计，我又掌握了许多新知识，例如 layUI、C#、.NET 等等，并且针对于数据库的联合查询等也可以运行的灵活自如，并且已经能够将技术理论与开发实践进行相结合。在最初的时候我在选择语言上面难以下手，但是经过查阅之后我发现适合自己的才是最好的。此次设计，对我以后的工作以及个人发展有着极大的益处。但是由于时间以及个人能力上的限制，本系统还存在着诸多问题，但核心功能部分已经全部实现。我相信随着我个人能力的提升，我会将其运用到真正需要的地方。

谢 辞

流光易逝、时光荏苒，大学四年一如白驹过隙匆匆过去，感谢我的指导老师，感谢我的学院、感谢我的学校，更要感谢我的舍友们，感谢所有人四年以来的陪伴与关怀，让我感受到了家的温暖。想起初入山东建筑大学，迈进大学，怀着惶恐与不安的心情面对着这诺大的校园。但是我知道，这里将是我人生新的起点，仍记得大一时面向对象编程技术让我打开了变成世界的新大门，等到了大二又开始了专业的新一轮学习与升级，到现在为止各种网络的配置安装一如昨日般浮现于我的脑海之中，等我迈入大三之后，面临着许许多多的计算机技术让我目不暇接，在其中挑选了自己比较喜欢的语言进行学习。到现在，马上面临毕业，即将进入职场的我已经能够从容因对各种总挑战,感谢各位老师的培养。

参考文献

- [1] Andrew Troelsen 著 朱晔 译 《C#与.NET 4 高级程序设计》2011 年 4 月
人民邮电出版社
- [2] 张越廷 顾彦玲 著 《ASP.NET 从入门到精通》2008 年 9 月 清华大学出版社
- [3] Stephen Walther 著 《ASP.NET 3.5 Unleashed》2008-01 Sams
- [4] 杨世锡 赵辉 《ASP+SQL Server 动态网站开发从基础到实践》第一版 电子工业出版社
- [5] 吴涛 《网站全程设计技术》 北京 清华大学出版社 北京交通大学出版社
- [6] 张华、贾志娟 《ASP 项目开发实践》 中国铁道出版社
- [7] 马伟 著 《ASP.NET 4 权威指南》2011 年 1 月 机械工业出版社
- [8] (美) Bill Evjen Scott Hanselman Devin Rader 著 《Professional ASP.NET 3.5 SPI Edition:In C# and VB》2009 年 4 月
- [9] Adam Freeman Steven Standerson 著 《Pro ASP.NET MVC 3 Framework》2012-5 Apress
- [10] Robin Dewen 著 董明 译 《SQL Server 2008 基础教程》2009 年 5 月
人民邮电出版社
- [11] 肖慧 基于网络教学平台的在线考试系统的开发[J] 科技导报 2018.15(11)
- [12] 单东林 张晓菲 魏然 著 《锋利的jQuery(第2版)》2012 年 7 月 人民邮电出版社