

Design and Implementation of Bird Collision Warning System for Airport

Kangli Zhang

Shandong Rural Credit Cooperatives Union, Jinan Shandong
Email: 1178982668@qq.com

Received: Feb. 9th, 2018; accepted: Feb. 21st, 2018; published: Feb. 28th, 2018

Abstract

Facing the increasingly serious situation of preventing bird and the economic loss increasing year by year, the author makes the airport bird strike warning and prevention system by combining the technology of high technology with ornithological knowledge. This system is applicable to the national military and civilian airports of different areas to prevent bird strike and ensure aviation safety. In this paper, a new database framework is developed in the design and implementation of the system, which reduces the instability and insecurity problems in the connection between the database and the system. The design target, system database and system implementation of the system are introduced in this paper. Finally, the author uses the system to make a statistical analysis of some birds' data. The design and implementation of the system can better serve the prevention of birds at the airport.

Keywords

Bird Striking, Avian Situation Alert, Database Framework

机场鸟撞预警防范系统的设计与实现

张康丽

山东省农村信用社联合社, 山东 济南
Email: 1178982668@qq.com

收稿日期: 2018年2月9日; 录用日期: 2018年2月21日; 发布日期: 2018年2月28日

摘要

面对日益严峻的防鸟形势和逐年增长的经济损失, 笔者将高科技技术和鸟类学知识相结合, 制作了“机

场鸟撞预警防范系统”，适用于全国大小、军民用机场，以此来防范鸟撞，保障航空安全。本文在系统的设计与实现中研发了新的数据库框架，减少了数据库与系统连接中不稳定与不安全问题的出现。文中介绍了系统设计目标、系统数据库、系统实现方式等。笔者最后利用该系统对部分鸟情数据进行统计分析。该系统的设计与实现可以更好地服务于机场对鸟类的防范。

关键词

鸟撞，鸟情预警，数据库框架

Copyright © 2018 by author and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

自 1912 年美国加州长滩发生海鸥撞击飞机引发的首起空难以来，鸟撞问题已成为影响世界航空安全的重要因素。不论是军用飞机还是民用飞机，鸟撞问题都是构成航空事故链中的重要一环，它已经成为航空界公认的严重致灾因素，并被国际航空联合会确定为 A 类安全灾难。

面对日渐复杂的空域危险，防范鸟撞、保障飞行安全成为国际航空界必须解决的科学难题。研究表明，飞机起降阶段是最容易遭受鸟类撞击的时段。对鸟情规律进行科学研究，加强科学驱鸟和对鸟情与鸟撞相关性的预测、预报研究，是有效降低鸟撞发生概率的重要方法。据统计，当前大多数鸟撞防范措施尚停留“驱赶”层面上[1] [2]，当发现可能马上来临的潜在危险鸟情时，地勤人员往往采取鸣枪示警、声音恐吓等传统方式驱赶；而在平时，地勤人员则采用气味驱赶、阻断生态链、恶化本地生态等方法将鸟类逼走。可以说，这些方法在短时间有一定作用，但其“被动驱赶”的特点使工作人员无法对鸟情进行提前预测，也就难以提前采取预防措施。

笔者针对上述情况，设计了“机场鸟撞预警防范系统”。该系统可以实时采取鸟情数据，对数据进行分析，得出鸟类规律变化，将鸟情信息提供给机场，从而及时采取鸟撞防范措施。

2. 系统的设计与实现

2.1. 系统设计目标

为填补鸟类管理与鸟情数据分析功能在全国范围内的技术空白，笔者以鸟类信息管理为主线，以鸟情分析为着力点，搭建了该系统。该系统以鸟类专家为技术支撑[3]，以各类鸟爱好者为大众基础，对鸟类信息资源进行整合，为机场提供鸟情方面的有力信息支持。

2.2. 系统数据库

机场鸟撞预警防范系统的数据来源主要有两种。

1) 通过该系统开放接口获取非标准化真实数据。该系统中具有在线百度地图，可以实时获取鸟类地理位置信息。

2) 鸟类活动的文献资料。该系统已将《山东鸟类分布名录》录入到数据库中。因此数据库中的信息都是真实有效的。如图 1 所示。



Figure 1. Bird information
图 1. 鸟情信息

2.3. 技术创新

2.3.1. NDolls 数据框架

NDolls 数据框架是该系统的特色。该框架是自行研制发明的，很好的解决了往常连接数据库存在的不稳定，不安全等问题。

开发者在进行系统项目开发的过程中，数据库是非要重要的方面。系统和数据库的连接经常出错。数据库连接出错，那么数据的安全就存在问题。目前常用的是三层框架，即 Model、View、Controller，这是软件设计中常用的标准。该三层框架在开发过程中，诸多问题。相互之间的依赖程度很高，如果某一层框架发生变化就会导致其它框架受到重大的影响。可谓是牵一发而动全身。这种情况下就加大了开发者的工作量，增加了工作的繁琐度，降低了工作效率，带来了不必要的麻烦，造成的是无用功[4] [5]。该三层框架针对表与表之间的关联处理的不够妥当，支持程度不高，一旦表的内容有所改变，一些原本已经完成的工作就需要重新做。

针对上述问题，提出了 NDolls 框架。NDolls 数据库框架可以解决目前数据库链接方面的弊端。NDolls 框架封装了底层数据库，这样就减少了直接操作底层数据库的麻烦。与此同时，对数据库进行操作时也一定程度上对底层数据库结构的依赖度降低。该框架并不是复杂难用而是简单可靠，它只是对多数据的支持通过简单的配置参数来实现。这样就拓展了 NDolls 框架的应用范围，不只是仅仅局限于一个数据库，而是多个，对 NDolls 数据库框架的推广也提供了极大的方便。目前，该 NDolls 数据库框架支持 SQL Server 数据库和 SQLite 数据库。随着该框架的使用和发展，在将来可以支持更多种类的数据库。NDolls 数据库框架也不再局限于.NET 平台。在未来的发展中可以发明更多的数据库框架，例如实现 java 语言与数据库的可靠连接，以此来减少程序开发人员的工作量。

目前研发的 NDolls 数据库框架需要在特定的环境中才能使用。该框架需要在 .Net Framework 4.0 及以上版本中使用。在项目中需要添加引用 NDoll.Core.dll 及 NDoll.Data.dll。

2.3.2. 数据库配置

开发人员可以根据具体的情况选择“代码方式”或者“配置文件方式”中的一种，来实现与数据库

的连接管理。目前，NDolls 框架中只有这两种实现方式。

SQL Server 和 SQ Lite 数据库都可以采用 NDolls 数据库框架。数据库类型的配置通过“DBType”实现，数据库连接的配置通过“Connection String”实现。DBtype 是数据库类型，包括 SQ Lite 和 SQL Server 数据库。Connection String 代表链接字符串。

配置文件配置方式中数据库连接的实现，要求 appSettings 中增加连接字符串项 Connection String。主项目的配置文件通过该解决方案来实现。配置文件配置方式实现代码如图 2 所示。

代码配置方式中数据库连接的实现需要加入代码进行配置。代码添加在没有系统调用数据库操作的位置即可。代码的配置方式实现如图 3 所示。

2.3.3. NDolls 常用功能

对数据库的操作不是完成几个配置就能实现操作的，在 NDolls 中对数据库操作的支持是通过建立容器来实现的。Repository 功能容器的创建可以实现操作数据库的目的。因此，只有先建立 Repository 功能容器，才可以操作数据库。该功能容器创建后，我们就可以利用容器的各种方法来实现数据库的增加、删除、修改、查询等常用的功能。容器的创建示例如图 4 所示。

在系统中手工录入数据，将这些数据新增到数据库中，需要调用 Add 方法，该方法是在 Repository 容器中写好的。底层的数据库通过 NDolls 自动实现插入(INSERT)操作。新增操作如图 5 所示。

系统中的数据进行修改时，这个时候就要调用 Update 方法，该方法已经在 Repository 容器中写好。底层数据库的修改操作在调用方法后可以通过 NDolls 自动实现。修改的操作如图 6 所示。

录入到数据库中数据需要保存在数据库中。Add Or Update 方法的调用能够实现数据的保存。该方法是封装在 Repository 容器中的。底层数据库的保存操作通过该 NDolls 自动实现。主键给 NDolls 提供了判断的标准。该数据库框架根据主键自动判断数据是修改还是增加的操作。保存操作如图 7 所示。

```
<appSettings>
  <add key="DBType" value="SQLite"/>
  <add key="ConnectionString" value="Data Source=|DataDirectory|App_Data/test.db;Pooling=True;Max
Pool Size=100"/>
</appSettings>
```

Figure 2. Configuration file configuration

图 2. 配置文件配置方式

```
NDolls.Data.DataConfig.ConnectionString = "Data Source=testdb;Pooling=False;Max Pool Size=100;";
NDolls.Data.DataConfig.DatabaseType = NDolls.Data.DBType.SQLite;
```

Figure 3. Code configuration

图 3. 代码配置方式

```
//创建针对 Model.Order 数据模型的 Repository 容器
public IRepository<Model.Order> r = RepositoryFactory<Model.Order>.CreateRepository("Model.Order");
```

Figure 4. Container creation example

图 4. 容器创建示例

```

//构造模型实例
Model.Order order = new Model.Order();
order.ID = ×××;
order.FlowNo = ×××;
order.GoodsNum = ×××;
order.OriginalPrice = ×××;
//将模型实例添加的数据库
if (r.Add(m))
{
    MessageBox.Show("添加成功");
}

```

Figure 5. Container creation example
图 5. 容器创建示例

```

//构造模型实例
Model.Order order = new Model.Order();
order.ID = ×××;
order.FlowNo = ×××;
order.GoodsNum = ×××;
order.OriginalPrice = ×××;
//将模型实例添加的数据库
if (r.Update(m))
{
    MessageBox.Show("添加成功");
}

```

Figure 6. Underlying database modification
图 6. 底层数据库修改

数据库的查询操作由查询项、条件项、排序项等组成。查询项方面，数据库的查询是由排序和条件来决定的。NDolls 抽象数据库查询过程中的因素为查询项。由于使用情况的不同，查询项又分为四种：Condition Item 类(条件项)、Order Item 类(排序项)、Condition And Group (与关系条件组)、Condition Or Group (或关系条件组)。如图 8 所示。

如果查询条件单一简单，没有复杂的组合条件，就使用条件项即可。条件项示例如图 9 所示。

条件项现在支持的条件查询有精确查询、模糊查询、左侧模糊查询、右侧模糊查询等。所有的查询如图 10 所示。

查询的排序方式通过排序项来定义查询。可以按升序查询也可以按降序进行查询。排序项示例如图 11 所示。

与关系条件组(Condition And Group)指的是多种条件的组合。与关系条件组示例如图 12 所示。

或关系条件组(Condition Or Group)指的是随机的条件的组合并非必须所有条件的组合。或关系条

```

//构造模型实例
Model.Order order = new Model.Order();
order.ID = xxx;
order.FlowNo = xxx;
order.GoodsNum = xxx;
order.OriginalPrice = xxx;
//将模型实例添加的数据库
if (r.AddOrUpdate(m))
{
    MessageBox.Show("保存成功");
}

```

Figure 7. The underlying database saves

图 7. 底层数据库保存

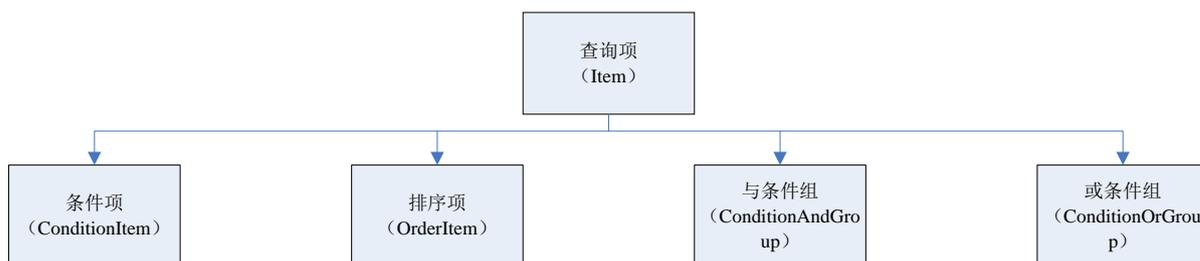


Figure 8. Composition of a query item

图 8. 查询项的组成

```

new ConditionItem("MemberNo", memberNo, SearchType.Accurate); //需 using NDolls.Data.Entity

```

Figure 9. Example term examples

图 9. 条件项示例

件组示例(User Name 包含 hello 或包含 kitty)如图 13 所示。

系统开发完成后进行查询时可以按照单个主键查询、多个主键组合查询、条件项查询。

按主键查询(单主键)示例如图 14 所示。

按主键查询(联合主键)示例如图 15 所示。

按条件项查询示例如图 16 所示。

2.4. 系统组成

系统包括管理前台和管理后台两部分。

2.4.1. 系统前台

系统前台主要包括：用户中心模块、鸟类查询模块、系统帮助模块最新动态。游客不用登陆可查询简单的鸟类信息。普通用户和管理员必须先注册，然后登陆。登陆后可以修改自己的注册信息，上传自己的拍客信息等。对于鸟类查询模块用户可以按照季节，按照名称，按照所处地域位置，按照鸟类颜色进行筛选查找。在按照名称查找时对于不常见的字我们可以采用拼音查找。系统帮助模块可以帮助用户

条件项值	条件项说明	备注
SearchType.Accurate	精确查询	FieldName='***'
SearchType.Fuzzy	模糊查询	FieldName LIKE '%***%'
SearchType.LeftFuzzy	左侧模糊查询	FieldName LIKE '%***'
SearchType.RightFuzzy	右侧模糊查询	FieldName LIKE '***%'
SearchType.UnContains	不包含	FieldName NOT LIKE '%***%'
SearchType.Unequal	不等于	FieldName < '***'
SearchType.ValuesIn	包含其中值	FieldName IN ('***','***')
SearchType.ValuesNotIn	不包含其中值	FieldName NOT IN ('***','***')
SearchType.Greater	大于	FieldName > ***
SearchType.Lower	小于	FieldName < ***
SearchType.GreaterEqual	大于等于	FieldName >= ***
SearchType.LowerEqual	小于等于	FieldName <= ***

Figure 10. Condition query example

图 10. 条件查询示例

```
new OrderItem("CreateTime", OrderType.ASC); //需 using NDolls.Data.Entity
```

Figure 11. Sort item example

图 11. 排序项示例

```
ConditionAndGroup g = new ConditionAndGroup();
ConditionOrGroup g1 = new ConditionOrGroup();
. . . . .
ConditionOrGroup g2 = new ConditionOrGroup();
. . . . .
g.AddCondition(g1);
g.AddCondition(g2);
```

Figure 12. Example of relation condition group

图 12. 关系条件组示例

```
ConditionOrGroup g = new ConditionOrGroup();
g.AddCondition(new ConditionItem("UserName", "hello", SearchType.Fuzzy));
g.AddCondition(new ConditionItem("UserName", "kitty", SearchType.Fuzzy));
```

Figure 13. Or example of a conditional group

图 13. 或关系条件组示例

尽快了解系统的一些基本操作方法和系统的功能[6] [7]。最新动态中包含最近的国内外研究成果，管理员推荐的热门论文及话题，用户可以就自己感兴趣的话题参与话题讨论。如图 17 所示。

2.4.2. 系统后台

管理后台是所有管理员管理系统数据针对系统运营的工具。目前该系统的后台采用.NET 框架，共分

```

IRepository<Model.Order> r;
r = RepositoryFactory<Model.Order>.CreateRepository("Model.Order");
Model.Order m = r.FindByPK("OrderID");

```

Figure 14. Main key query example

图 14. 主键查询示例

```

IRepository<Model. SupplierGoods> r = RepositoryFactory<Model.
SupplierGoods>.CreateRepository("Model.SupplierGoods");
Model.SupplierGoods m = r.FindByPK(new string[] { "SID", "GoodsID" });

```

Figure 15. Example of the joint primary key query

图 15. 联合主键查询示例

```

IRepository<Model.Order> r;
r = RepositoryFactory<Model.Order>.CreateRepository("Model.Order");
List<Item> conditions = new List<Item>();
ConditionItem c1 = new ConditionItem("MemberNo", memberNo, SearchType.Accurate);
ConditionItem c2 = new ConditionItem("CreateTime", '2015-12-30', SearchType.Greater);
conditions.Add(c1);
conditions.Add(c2);
var list = r.FindByCondition(conditions);

```

Figure 16. Condition item query example

图 16. 条件项查询示例

为鸟类信息、日常运营、会员管理、系统设置、帮助五大功能模块，实现系统所有信息的管理。如图 18 所示。

1) 鸟类信息

鸟类知识库中已经将鸟类的信息导入到数据库。后台管理员可以通过该平台对鸟类的静态信息，例如鸟类的颜色、形状大小、分布区域、出现的季节、有无危害性等进行添加。该平台也可以对鸟类的动态信息进行添加[8]，在该模块中加载了地图，可以从地图上选择经纬度，从而确定鸟类的分布地点，这样更加准确的保存了鸟类的动态信息。该功能都已经实现。如图 19 所示。

2) 日常运营

子模块包括文章管理、日志查询。其中的文章根据具体需要实现了树形分类管理，用户登录后可以实现所有文章的“增删改查导”管理；日志查询，主要是结合前台用户登录后的具体操作信息进行的记录，后台也具备了“增删改查导”的功能。如图 20 所示。

3) 会员管理

子模块分为普通用户、专家用户、管理员。

其中普通用户是各类分支用户的基础，专家用户是在普通用户基础上的“拓展”，后台具备了这三类用户的“增删改查导”功能。如图 21 所示。

4) 系统设置

包括字典库管理、国家地点管理、文章类别管理、鸟类类型管理、用户角色管理、用户权限管理。

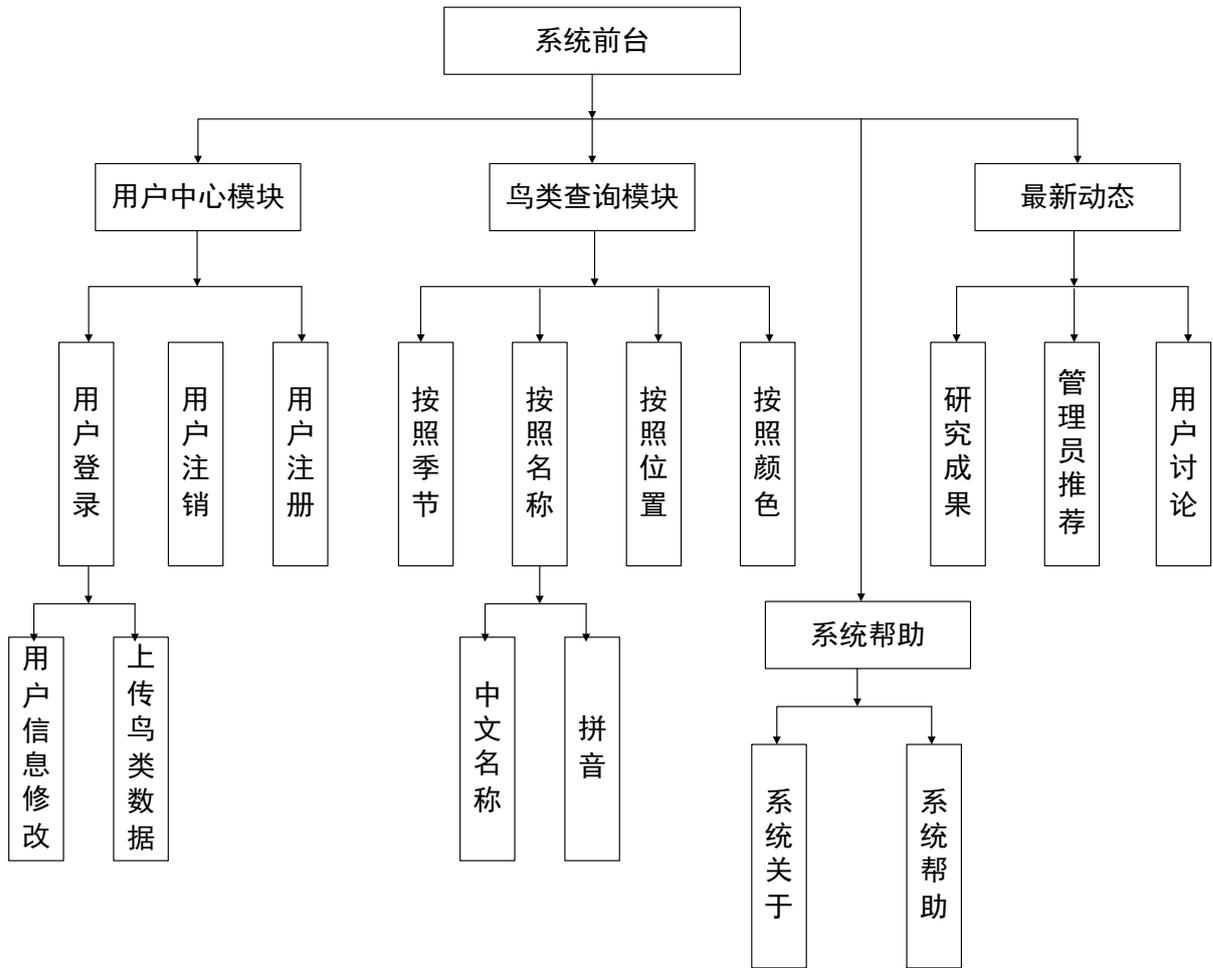


Figure 17. Loading map page
图 17. 加载地图页面

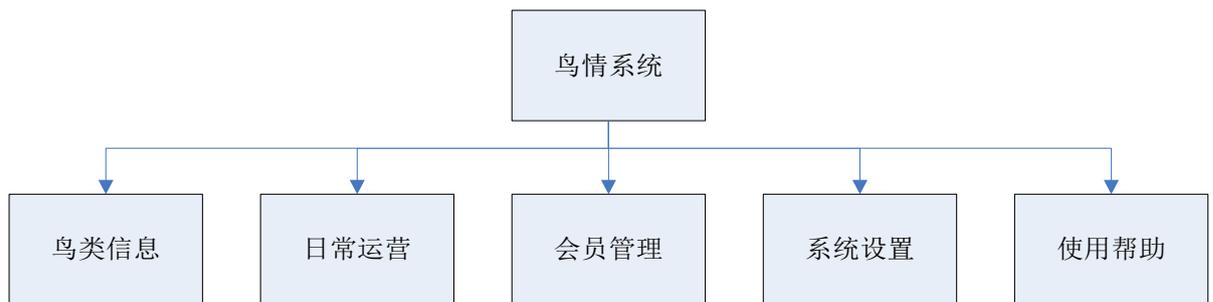


Figure 18. System functional structure diagram
图 18. 系统功能结构图

该功能仅对个别系统管理员开放，其中的字典库为系统内所有分类(1级分类)提供技术支持；国家地点管理将全国行政范围以“省市区街居”为层次进行划分和管理[9]；文章类别管理是根据鸟类文章的具体需要实现的树形管理；鸟类类型管理是根据鸟的科目属实现的分层管理；用户角色管理是针对各类管理员登陆后台后，各自具备的操作权限而设计的，每种角色具备与众不同的权限，通过管理员的 Role 记录各自的角色。如图 22 所示。

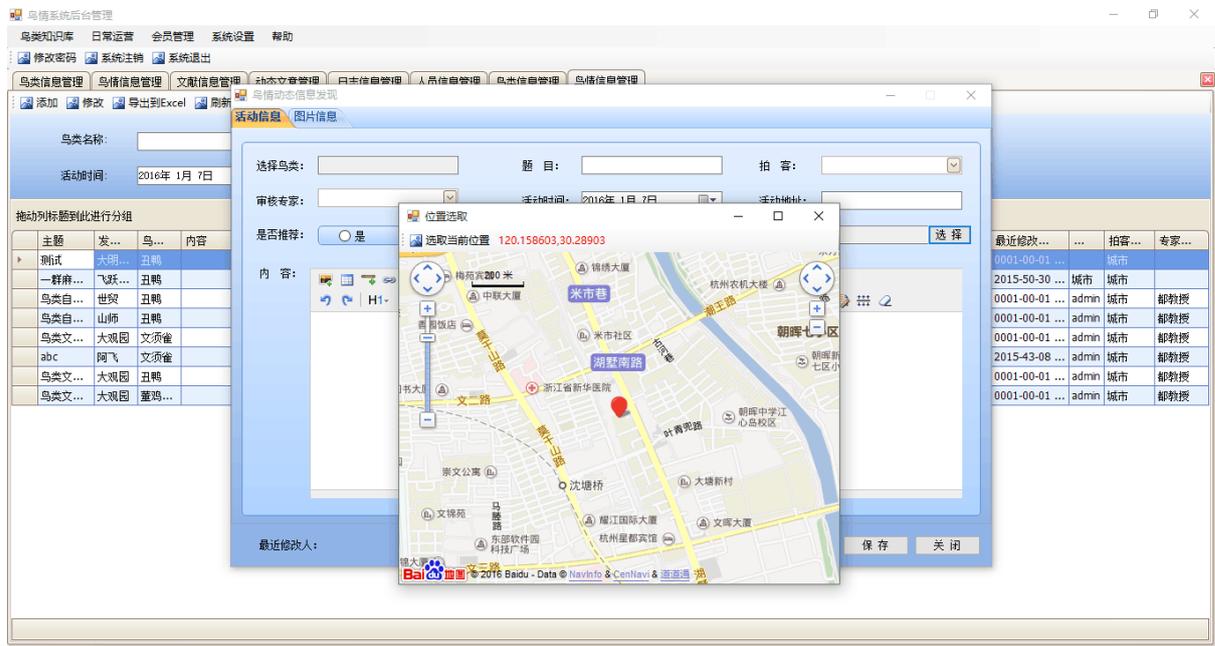


Figure 19. Loading map page

图 19. 加载地图页面

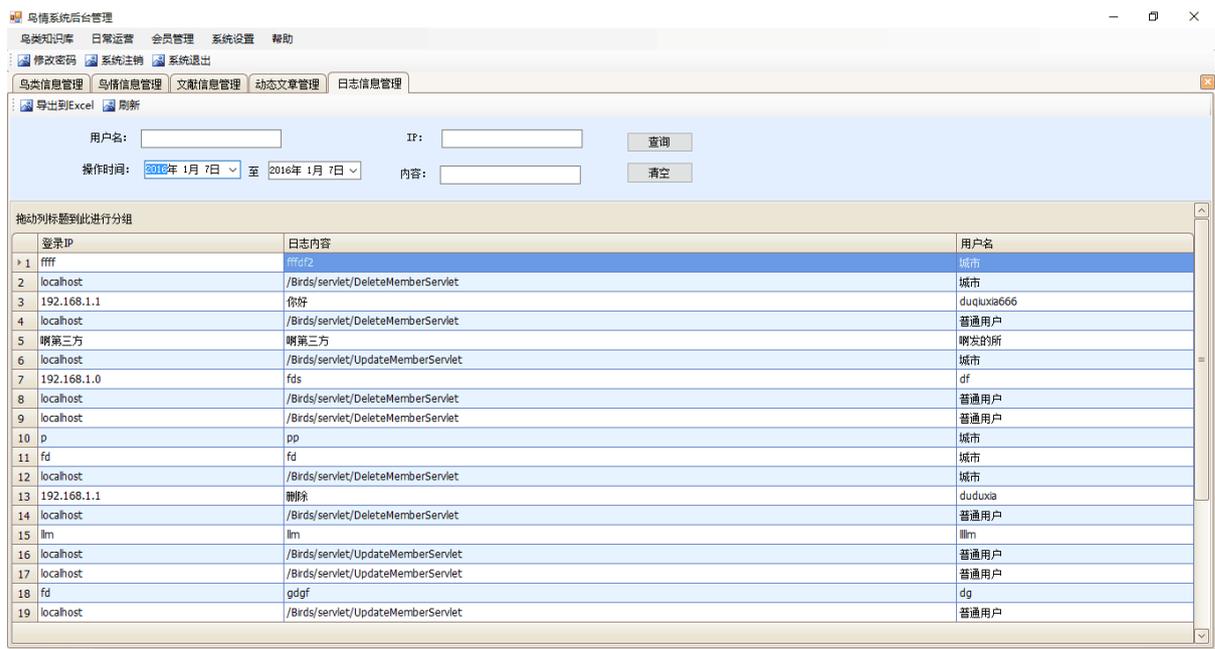


Figure 20. Daily operation

图 20. 日常运营

2.5. 系统实现

本系统采用基于.NET 系统开发, 使用 C#编程语言, 初期采用 SQL Server 数据库(后期视情况切换到 Oracle 数据库), 采取基于 NDolls 的多层体系架构实现。地图展示方面, 使用互联网开放的百度地图 API(需联网支持)实现各种效果[10]。



Figure 21. Member management
图 21. 会员管理



Figure 22. System settings
图 22. 系统设置

2.6. 鸟情数据统计分析

鸟情研究在社会生活中发挥着越来越重要的作用。目前，通过该系统采集的数据形式多样，来源广泛。在本文中抽出机场鸟情方面的数据，进行分析。为鸟撞防范提供真实的理论基础。

对采集到的数据按照天气的不同进行统计分析。如果天气寒冷，春天的时候，候鸟到北方来的时间

会推迟，秋天的季节，鸟类会提前飞往南方。天气的变化对鸟类的影响极大，在某种程度上改变了鸟类的栖息地。夏天和秋天的季节，如果有暴风雨的天气来临，鸟类会因为捕捉机场周围田地上的昆虫而聚集到一起。大雨过后，空气清新，一些蚯蚓、蜗牛等会到地面上，这些动物吸引了大批鸟类来觅食。下雪过后，清扫机场，鸟类常常选择这里的空地为活动场所。这就加大了鸟撞事故的发生。天气的变化影响了航空路线[11]，也影响了鸟类的活动规律，加大了鸟撞防范的难度。

该系统建立了鸟情数据库，既有文献资料也有拍客、机场人员上传的现实数据。系统中观察记录的内容包括鸟类名称、发现地点、活动时间、出现季节、外观、危险性、活动地址。系统内部有统计分析机制，如果使用人员想要统计出在某时间段出现的鸟类[12]，则可以采用查询的方式得到。例如：统计分析出 2014.7.22~2016.1.7 活动时间段的鸟类名称，以及它们的活动地点。统计出该信息之后，则在机场旁边的这些地点进行重点驱赶对应的鸟类。在这种情况下，相关鸟类就会减少了鸟撞事故的发生。统计分析如图 23 所示。

通过鸟情数据的统计分析，机场使用人员对鸟类的活动规律才能进行初步的掌握。机场使用人员掌握了这些科学的数据，就会与鸟撞规律进行结合[13]，主动使用正确的驱鸟措施，极大的减少了鸟撞的风险。该系统对机场使用人员录入的数据进行了统计分析，数据具有非常大的现实意义。对鸟情数据的分析，可以得出鸟类的分布规律，季节变迁，数量变化，从而有针对性地进行驱鸟，胜算更大。该系统的设计与实现为人工智能驱鸟的研究提供了科学依据。

该系统的使用人员对机场的几种鸟一天里的活动时间、出现的数量进行了记录。统计分析结果可以通过 EXCEL 表格导出。如图 24 所示。

对采集到的鸟情数据按照鸟类的分布地点进行统计分析。可以得出机场周边地区固定的鸟类。机场驱鸟人员可以根据该科学统计数据[14]，找到针对这几种鸟类的防范措施。很大程度上减少了鸟撞事故的发生。机场的周围有很多农田、森林、河流湖泊等。鸟类在这一带地区出现的频率较高。随着气候的变化，季节的变迁，鸟类活动也会有明显的周期变化[15]。该系统可以针对每种鸟的季节性变化进行统计分

The screenshot shows a web-based interface for bird information management. It includes search filters for bird name, location, and activity time. Below the filters is a table with columns for subject, author, bird name, content, activity time, creation date, last update, and user roles (photographer and expert).

主题	发...	鸟...	内容	活动时间	生成日期	最近修改...	...	拍客...	专家...
测试	大明...	丑鸭		2015-00-02 ...	2015-17-06 ...	0001-00-01 ...		城市	
一群麻...	飞跃...	丑鸭	一群麻雀，叽叽喳喳的	2015-00-30 ...	2015-44-30 ...	2015-50-30 ...		城市	
鸟类自...	世突	丑鸭		2015-00-22 ...	2015-19-22 ...	0001-00-01 ...	admin	城市	都教授
鸟类自...	山师	丑鸭		2015-00-22 ...	2015-19-22 ...	0001-00-01 ...	admin	城市	都教授
鸟类文...	大观园	文须雀		2015-00-22 ...	2015-07-22 ...	0001-00-01 ...	admin	城市	都教授
abc	阿飞	文须雀	agcd啊的发送的啊	2015-00-11 ...	2015-48-11 ...	2015-43-08 ...	admin	城市	都教授
鸟类文...	大观园	丑鸭		2015-00-22 ...	2015-07-22 ...	0001-00-01 ...	admin	城市	都教授
鸟类文...	大观园	董鸡...		2015-00-22 ...	2015-05-22 ...	0001-00-01 ...	admin	城市	都教授

Figure 23. Statistical analysis interface for the time period of bird activity

图 23. 鸟类活动时间段统计分析界面

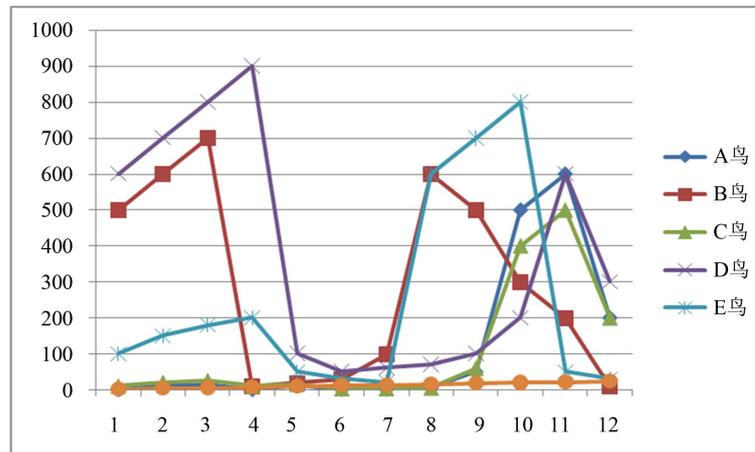


Figure 24. The date activity curve of several kinds of birds at an airport
图 24. 某机场几种鸟的日期活动曲线

析。通过查询的功能可以统计分析出某天鸟类的活动，数量的多少。从而根据这些数据有效地进行驱鸟。

3. 系统研发展望

- 1) 系统后期升级中将鸟情信息通过地图方式展示出来，具备电子地图基本查询功能；界面友好。
- 2) 在系统对新加入的鸟类资料，要进行筛选，识别等工作，最终并入鸟类分布数据库。
- 3) 系统基本完善之后。考虑采用数据挖掘算法、机器学习算法、智能算法对数据进行研究。选择并固定特定区域，给出该区域多年来、近年来、近期一段时间的鸟情活动、变迁规律。
- 4) 选择并固定特种鸟类，给出该鸟类在各区域的多年来、近年来，近期一段时间的活动规律；给出鸟类的种群活动、发展趋势预测；给出鸟类危害等级预测。
- 5) 给出未来一年、半年、3个月、1个月、15天、7天，3天，1天内的鸟情预测。

参考文献 (References)

- [1] 赛道建, 胡堃, 刘建. 济南城市繁殖鸟类生境选择研究[C]//中国动物学会鸟类学分会. 第八届中国动物学会鸟类学分会全国代表大会暨第六届海峡两岸鸟类学研讨会论文集: 2005 年卷. 济南: 中国动物学会鸟类学分会, 2005: 3.
- [2] Richard, D., Edward, J. and Timothy, R. (2010) Adding Attributes to Role-Based Access Control. *IEEE Computer Society*, 6, 79-81.
- [3] 熊高. 基于移动终端的会议管理系统设计与实现[D]: [硕士学位论文]. 武汉: 华中师范大学, 2013.
- [4] 张昌进. 基于 Multi-Agent 的移动个性化搜索关键技术研究[D]: [硕士学位论文]. 长沙: 湖南大学, 2013.
- [5] 陈成. 机场鸟类动态与防治研究[D]: [硕士学位论文]. 保定: 河北农业大学, 2013.
- [6] 田娟. 学校教职工体检管理信息系统的设计与实现[D]: [硕士学位论文]. 成都: 电子科技大学, 2014.
- [7] 李立. 接触网鸟害防治措施的探讨[J]. 价值工程, 2015(3): 94-96.
- [8] 周铭钰. 防范鸟击“大数据”来助阵[N]. 中国民航报, 2016-09-19(006).
- [9] 蒋裕. 浅谈我国鸟情对民机飞行安全的影响[J]. 科技视界, 2016(18): 248, 259.
- [10] 王增年, 施泽勇, 刘菊娥, 王旭春, 李晓云, 杨美. 机场鸟情预测预报系统论证初步框架[J]. 北京农业, 2016(2): 202-207.
- [11] 赛道建, 孙涛. 鸟撞防范概论[M]. 北京: 科学出版社, 2012: 46-72.
- [12] 张振飞. 机场鸟击防范措施研究[J]. 科技风, 2017(13): 243.
- [13] 蔡恒强. MVC 模式下餐饮管理系统设计与实现[J]. 电脑知识与技术, 2013(6): 1359-1361.

-
- [14] 余洋. 山东省农村支付环境建设信息化管理系统的研究与实现[D]: [硕士学位论文]. 济南: 山东大学, 2014.
- [15] 张黎明, 涂鹏, 吴炜, 陈彦, 曾小舟. 南昌青云谱机场鸟击防范措施研究[J]. 西南师范大学学报(自然科学版), 2017(2): 141-147.

知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2325-2286, 即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: sea@hanspub.org