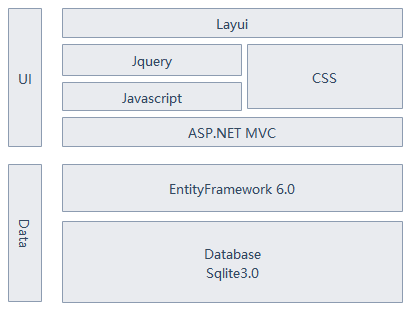
# 需求分析

本系统分析如下表：

|  |  |
| --- | --- |
| 需求 | 分析 |
| （1） 提供基于WEB的管理系统，支持Chrome、Firefox、IE10及以上； | 选择一个支持这些浏览器的前端界面框架即可 |
| （2） 提供用户管理及权限验证功能 | 设计权限系统即可 |
| （3） 提供管理员操作审计功能，支持对审计数据进行查询和统计； | 设计审计管理和查询模块 |
| （4） 跟踪数据库所有变更历史，记录数据库变更事件、变更时间、变更内容、操作人，包括数据、表、触发器、存储过程等； | 对要跟踪的数据库对象开启对象审计项；  采集操作对象的审计记录，保存在变更记录表中 |
| （5） 支持数据库记录变更比对功能； | 根据两个时间点从变更记录表中取数据来对比 |
| （6） 支持查询任意时刻数据库的状态，支持导出任意时刻数据库的完整镜像； | Oracle使用expdp工具  达梦使用dexp工具 |
| （7） 移除指定历史变更记录； | 从删除变更记录表中删除 |
| （8） 开启、关闭数据库变更监控； | 设计监控开头模块 |
| （9）支持的目标数据库：达梦数据库DM7、Oracle 10g/11g； | 针对具体数据库进行开发 |
| （10） 可配置的创建、修改、删除监控策略； | 设计监控策略管理模块 |

# 系统设计

本系统架构如下图：



## UI设计

系统UI采用了目前国内较流行的开源前端 UI 框架——Layui。layui（谐音：类UI) 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。layui 兼容人类正在使用的全部浏览器（IE6/7除外），可作为 PC 端后台系统与前台界面的速成开发方案。

## 后台框架

系统后台采用微软的ASP.NET MVC5框架。MVC模式将构成一个人机交互应用涉及的功能分为Model、Controller和View三部分，将关注点分离，使得开发与维护都方便许多。

## 数据库访问

数据库的访问采用了微软的Entity Framework（以下简称EF）作为系统的ORM框架，它目前支持多表映射，支持多种主流数据库。使用EF后，软件开发过程直接操作对象进行数据存储，避免了大量使用SQL语句，减少了系统开发的复杂性，同时也提高了系统的稳定性。

## 数据仓库

数据库使用了关系型的数据库SQLite3.0。SQLite作为一款轻型的数据库，不需要安装即可使用，有着占用资源低、支持主流的操作系统、处理速度快等优点。

# 功能模块

# 数据结构