

РАЗРАБОТКА ПРИЛОЖЕНИЯ УЧЕТА ОБОРУДОВАНИЯ В БАНКОВСКОЙ СИСТЕМЕ

Дансаранов С.В., Федорова К.А.
Томский политехнический университет
634050, Россия, г. Томск, пр-т Ленина, 30
E-mail: DansS@bk.ru

Введение

В настоящее время происходит повсеместное активное внедрение информационных технологий в различные структуры и организации. Компьютерные системы нужны везде, во всех сферах общественной жизни. Они призваны решать насущные проблемы, автоматизировать и упрощать работу человека.

Разрабатываемое приложение нацелено на автоматизацию учета оборудования в банке, где каждая единица техники прикреплена к конкретному сотруднику и отделу. В течение длительного времени на предприятии не велся учет техники, оборудование переходило от одного сотрудника к другому, из отдела – в отдел. И на данный момент не существует единого источника информации о состоянии и расположении оборудования.

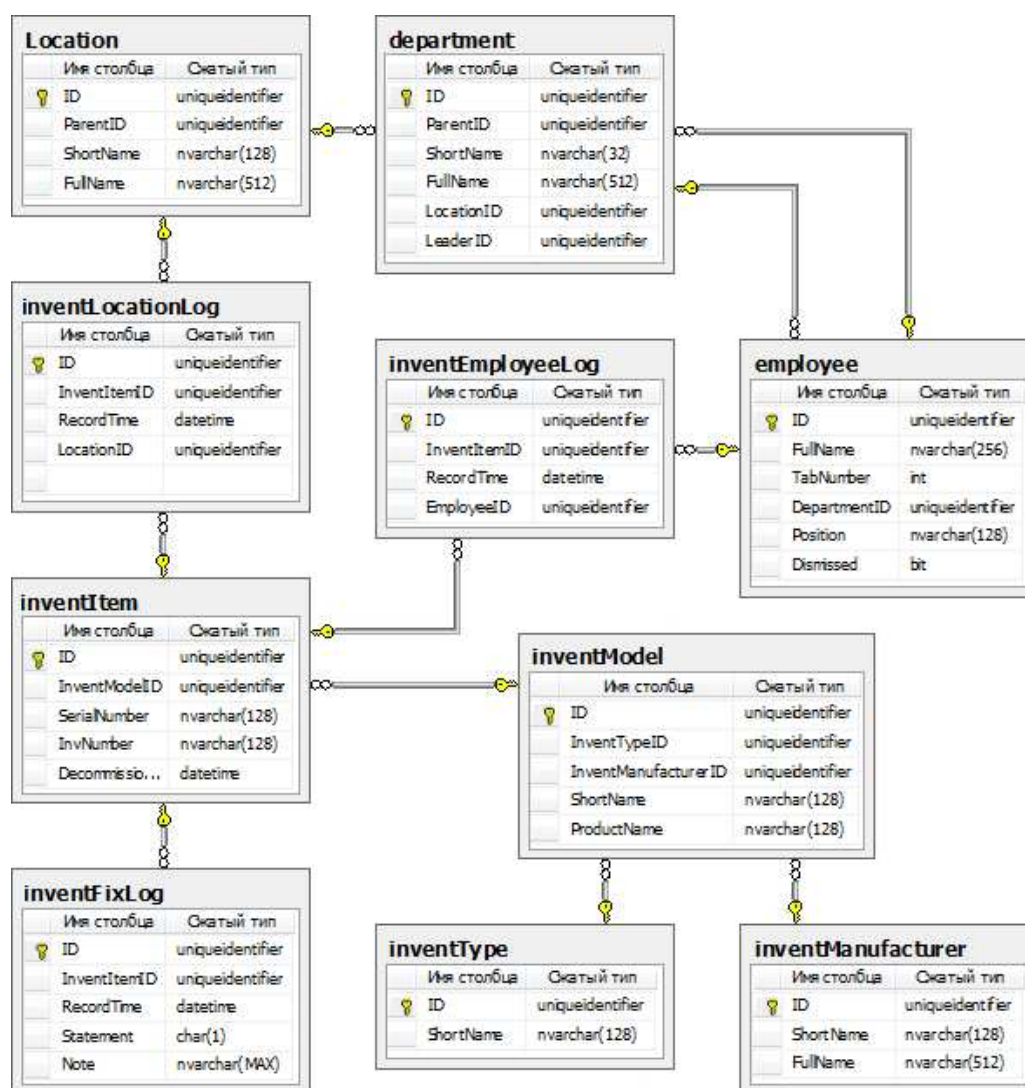


Рис. 1. Диаграмма базы данных

Выбор программных средств

Для разработки программного продукта была выбрана технология ASP.NET, в частности архитектурный паттерн MVC (Model – View – Controller). Шаблон проектирования MVC предполагает разделение данных приложения, пользо-

вательского интерфейса и управляющей логики на три отдельных компонента: Модель, Представление и Контроллер – таким образом, что модификация каждого компонента может осуществляться независимо. Модель служит для извлечения и манипуляций данными приложения, Представление

отвечает за видимое пользователю отображение этих данных, а Контроллер предоставляет взаимодействие предыдущих компонентов [1]. Приложение разрабатывалось в программной среде Microsoft Visual Studio 2010. Для создания базы данных использовалась среда Microsoft SQL Server Management Studio.

Создание базы данных

База данных должна отвечать требованиям, которые предъявлялись к программе, а именно:

- привязка каждой единицы оборудования к конкретному сотруднику и отделу;
- переназначение единицы оборудования к вышестоящему сотруднику в случае увольнения нижестоящего;
- отслеживание перемещения оборудования и его состояния (исправен, частично неисправен, неисправен);
- ведение истории всех перемещений и изменений состояния единицы оборудования.

На рисунке 1 представлена диаграмма базы данных, которая отражает основные характеристики объектов, а также зависимости между ними.

Разрывы связей на рисунке 1 обозначены соответствующими буквами.

Ниже описано содержание каждой таблицы:

- 1) department – информация об отделах предприятия;
- 2) employee – данные о сотрудниках предприятия;
- 3) location – информация о размещении оборудования (адрес, кабинет и т.п.);
- 4) inventModel – данные о моделях оборудования;
- 5) inventManufacturer – информация о производителях оборудования;
- 6) inventType – информация о типе оборудования (например монитор, системный блок);
- 7) inventItem – данные о каждой единице оборудования;
- 8) inventLocationLog – история изменений размещения оборудования;
- 9) inventEmployeeLog – история изменений привязки оборудования к сотруднику;
- 10) inventFixLog – журнал ремонта оборудования.

Разработка компонентов приложения

Для каждой таблицы базы данных необходимо создать свой класс, называемый моделью, где прописаны методы и поля ему присущие, т.к. разработка ведется с помощью шаблона MVC. Также был создан вспомогательный класс Tools, в котором находятся наиболее часто вызываемые функции некоторых классов.

В данном приложении для каждой модели создан свой контроллер, обеспечивающий связь между пользователем и системой, а также HomeController для отображения домашней страницы.

Представление отвечает за отображение данных пользователю, реагируя на изменения модели. Часто в качестве представления выступает форма (окно) с графическими элементами [2].

Необходимо отметить, что представление и контроллер зависят от модели, а модель, в свою очередь, от этих компонентов не зависит. Таким образом, достигается разделение функций, позволяющее строить модель независимо от визуального представления, а также создавать несколько различных представлений для одной модели.

Демонстрация работы приложения

При запуске программы открывается домашняя страница, где перечислены доступные ссылки. В левой части окна находятся основные страницы приложения.

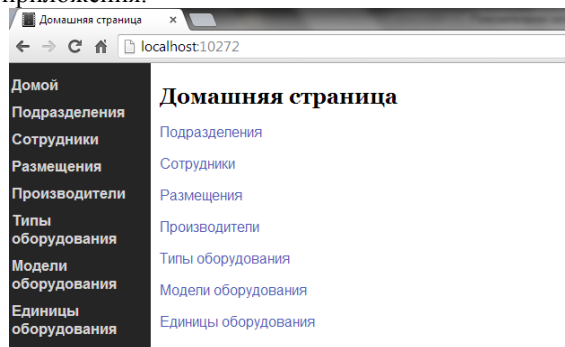


Рис. 2. Домашняя страница приложения

Таким образом, на каждой странице отображена информация из соответствующих таблиц в базе данных.

Заключение

В результате проделанной работы создано веб-приложение, являющееся хорошим фундаментом для разработки конечного программного продукта. В дальнейшем планируется расширять функционал программы путем добавления новых классов и усовершенствования уже имеющихся. Также будет улучшен интерфейс программы и ее внешнее представление.

Литература

1. ASP.NET MVC 3 в деталях – режим доступа: <http://andrey.moveax.ru/mvc3-in-depth/>
2. Учебный курс по ASP.NET MVC 3 – режим доступа: <http://blogs.msdn.com/b/ruwebapps/archive/2011/11/16/asp-net-mvc-3-4.aspx>