Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Модели данных и системы управления базами данных (МДиСУБД)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту на тему

«База данных сервиса по сдаче квартир на сутки»

Выполнил: студент гр.253504

Волчецкий А.М.

Руководитель: ассистент кафедры информатики Бугара Д.А.

**Оглавление**

[Введение 3](#__RefHeading___Toc947_524480692)

[1.Постановка задачи 5](#__RefHeading___Toc949_524480692)

[1.1.Описание предметной области 5](#__RefHeading___Toc951_524480692)

[1.2.Задачи проектирования 5](#__RefHeading___Toc953_524480692)

[1.3.Требования к разрабатываемому программному обеспечению 6](#__RefHeading___Toc955_524480692)

[2. Проектирование задачи 7](#__RefHeading___Toc957_524480692)

[2.1.Проектирование схемы базы данных 7](#__RefHeading___Toc959_524480692)

[2.2.Используемые технологии 7](#__RefHeading___Toc961_524480692)

[2.3.Среды реализации задачи 8](#__RefHeading___Toc963_524480692)

[2.3.1.Microsoft Visual Studio 8](#__RefHeading___Toc965_524480692)

[2.3.2.C# 8](#__RefHeading___Toc967_524480692)

[2.3.3.ASP.NET MC 9](#__RefHeading___Toc969_524480692)

[2.3.4.MSSQL 10](#__RefHeading___Toc971_524480692)

[3. Структура приложения 11](#__RefHeading___Toc973_524480692)

[4. Тестирование 12](#__RefHeading___Toc975_524480692)

[5. Описание применения 14](#__RefHeading___Toc977_524480692)

[Заключение 15](#__RefHeading___Toc979_524480692)

[Список использованных источников 16](#__RefHeading___Toc981_524480692)

# Введение

База данных – это совокупность структур, предназначенных для хранения больших объемов информации и программных модулей, осуществляющих управление данными, их выборку, сортировку и другие подобные действия.

Информация базы данных хранится в одной или нескольких таблицах. Любая таблица с данными состоит из набора однотипных записей, расположенных друг за другом. Они представляют собой строки таблицы, которые можно добавлять, удалять или изменять.

Каждая запись является набором именованных полей, или ячеек, которые могут хранить самую разнообразную информацию, начиная от даты рождения и заканчивая подробным описанием кулинарного рецепта. Однотипные поля разных записей образуют столбец таблицы.

Создав одну таблицу, вы уже получаете полноценную базу данных. Однако в реальной жизни структуры баз данных, а соответственно и способы их создания, намного сложнее.

В последние годы на первый план выдвигается новая отрасль - информационная индустрия, связанная с производством технических средств, методов, технологий для производства новых знаний. Эта индустрия тесно связана с развитием компьютерных технологий.

В информационном обществе доминирует производство информационного продукта, а материальный продукт становится более информационно емким. Изменятся весь уклад жизни, система ценностей: возрастает значимость культурного досуга, возрастает спрос на знания, от человека требуется способность к интеллектуальному труду и творчеству. В результате появились противоречия между ограниченными возможностями человека по восприятию и переработке информации и существующими массивами хранящейся и передаваемой информации.

Возникло большое число избыточной информации, в которой иногда трудно сориентироваться и выбрать нужные сведения.

Для решения подобных проблем применяются автоматизированные базы данных. Они стали неотъемлемой частью практически всех компьютерных систем - от отрасли до отдельного предприятия. За последние несколько лет вырос уровень потребительских качеств систем управления базами данных (СУБД): разнообразие поддерживаемых функций, удобный для пользователя интерфейс, сопряжение с программными продуктами, в частности с другими СУБД, возможности для работы в сети и т.д. СУБД позволяет сводить воедино информацию из самых разных источников (электронные таблицы, другие базы данных) и помогает быстро найти необходимую информацию, донести ее до окружающих с помощью отчетов, графиков или таблиц.

К настоящему времени накоплен значительный опыт проектирования БД, предназначенных для управления производством, это позволяет сделать процесс создания БД более эффективным.

# Постановка задачи

## Описание предметной области

Областью применения данной базы данных является сервис по сдаче квартир на сутки. Администраторы могут легко добавлять новые квартиры с информацией о них, а обычные пользователи могут легко их арендовать.

## Задачи проектирования

Чтобы приступить к непосредственному проектированию базы данных, необходимо выделить минимальный набор основных сущностей и указать типы связи между ними.

В рассматриваемой предметной области можно выделить следующие основные сущности:

Клиент – содержит информацию о пользователе и его статус

Квартира — содержит информацию о квартире, количество комнат, стоимость

Типы связей между сущностями.

**1:n**:

AFYCustomer – AFYProductOrder,

AFYCustomer – Product,

City – Country

**m:n**:

AFYCustomer– AFYCustomerRole.

## Требования к разрабатываемому программному обеспечению

Готовый программный продукт должен удовлетворять следующим требованиям:

1. Иметь простой и понятный интерфейс;
2. Доступ к базе данных через приложение должен осуществляться посредством технологии Entity Framework;
3. СУБД: MS SQL Server;
4. Позволять легко добавлять/удалять данные о клиентах;
5. Предоставлять актуальную информацию о свободных квартирах;
6. Иметь возможность найти квартиру;

Таким образом, задача данного курсового проекта сводится к разработке приложения для обеспечения работы сервиса по сдаче квартир. Готовый сайт должно иметь понятный интерфейс и быть удобным в использовании.

# Проектирование задачи

## Проектирование схемы базы данных

Для разработки базы данных была использована СУБД MSSQL.

На рис.1 представлена схема спроектированной базы данных. На ней можно видеть сущности, типы связей и атрибуты.

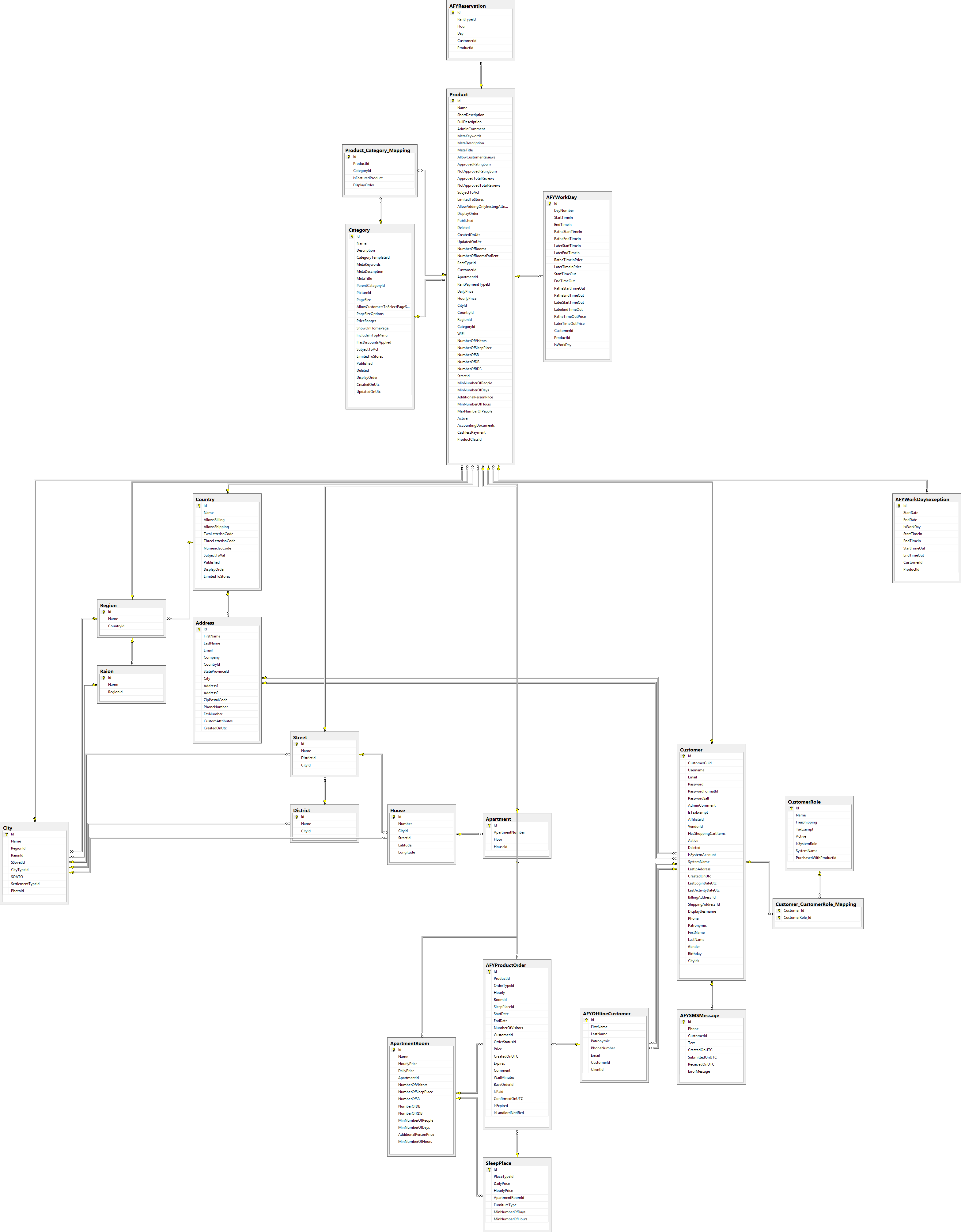


Рис. 1. Схема базы данных

## Используемые технологии

Приложение реализовано на языке C# с использованием технологии ASP.NET MVC и Entity Framework. В качестве среды разработки была выбрана Microsoft Visual Studio 2015. Для создания и работы с базой данных была использована СУБД MSSQL.

## Среды реализации задачи

### Microsoft Visual Studio

Microsoft Visual Studio – это новая разработка компании Microsoft, позволяющая создавать приложения, работающие на платформе .NET. Особенность этой платформы заключается в широком наборе сервисов, которые доступны в различных языках программирования. При этом сервисы реализуются в виде промежуточного кода, который не зависит от базовой архитектуры. Едва ли не главной целью создания такой платформы было оснащение разработчиков специальными сервисно-ориентированными приложениями, которые могли бы работать на любой платформе, начиная от персонального компьютера и заканчивая мобильным устройством.

Microsoft Visual Studio объединяет в себе огромное количество функций, позволяющих осуществлять разработки для Windows всех версий, в том числе и Windows 8, интернета, SharePoint, различных мобильных устройств и облачных технологий. В Visual Studio реализуется новая среда разработчика, благодаря которой создавать приложения стало проще. Microsoft Visual Studio – это обновленная и упрощенная программная среда, для которой характерна высокая производительность, причем она не зависит от особенностей оборудования.

### C#

C# разрабатывался как язык программирования прикладного уровня для CLR и, как таковой, зависит, прежде всего, от возможностей самой CLR. Это касается, прежде всего, системы типов C#, которая отражает BCL. Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR. Так, с развитием CLR от версии 1.1 к 2.0 значительно обогатился и сам C#; подобного взаимодействия следует ожидать и в дальнейшем. (Однако эта закономерность была нарушена с выходом C# 3.0, представляющего собой расширения языка, не опирающиеся на расширения платформы .NET.) CLR предоставляет C#, как и всем другим .NET-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования. Например, сборка мусора не реализована в самом C#, а производится CLR для программ, написанных на C# точно так же, как это делается для программ на VB.NET, J# и др.

### ASP.NET MVC

ASP.NET MVC представляет собой платформу для создания сайтов и веб-приложений с использованием паттерна (или шаблона) MVC (model - view - controller).

Работа над новой платформой была начата в 2007 году, а в 2009 году появилась первая версия. В итоге к текущему моменту (2012 год) уже было выпущено 4 версии платформы, а сам фреймворк обрел большую популярность по всему миру благодаря своей гибкости и адаптивности.

Шаблон MVC, лежащий в основе новой платформы, подразумевает взаимодействие трех компонентов: контроллера (controller), модели (model) и представления (view). Что же представляют эти компоненты?

**Контроллер** (controller) представляет класс, с которого собственно и начинается работа приложения. Этот класс обеспечивает связь между моделью и представлением. Получая вводимые пользователем данные, контроллер исходя из внутренней логики при необходимости обращается к модели и генерирует соответствующее представление.

**Представление** (view) - это собственно визуальная часть или пользовательский интерфейс приложения - например, html-страница, через которую пользователь, зашедший на сайт, взаимодействует с веб-приложением.

**Модель** (model) представляет набор классов, описывающих логику используемых данных.

Общую схему взаимодействия упрощенно можно представить следующим образом:

### ASP.NET MVC и ASP.NET Web Forms

ASP.NET MVC является в некотором роде конкурентом для традиционных веб-форм и имеет по сравнению с ними следующие преимущества:

* **Разделение ответственности**. В MVC приложение состоит из трех частей: контроллера, представления и модели, каждая из которых выполняет свои специфичные функции. В итоге приложение будет легче поддерживать модифицировать в будущем.
* В силу разделения ответственности приложения mvc обладают **лучшей тестируемостью**. И мы можем тестировать отдельные компоненты независимо друг от друга.
* **Соответствие протоколу HTTP**. Приложения MVC в отличие от веб-форм не поддерживают объекты состояния (ViewState). Ясность и простота платформы позволяют добиться большего контроля над работой приложения
* **Гибкость**. Вы можете настраивать различные компоненты платформы по своему усмотрению. Изменять какие-либо части конвейера работы MVC или адаптировать его к своим нуждам и потребностям.

В то же время не стоит однозначно сбрасывать со счетов ASP.NET WebForms. Поскольку она также имеет свои сильные стороны, например, модель событий, которая будет ближе тем разработчикам, которые ранее занимались созданием клиентских приложений.

В традиционных веб-формах вы имеете контроль над разметкой и можете в реальном времени в визуальном редакторе Visual Studio увидеть, как будет выглядеть та или иная страница. При работе с MVC Visual Studio подобного не позволяет делать.

В любом случае вы вольны выбирать ту платформу, которая приходится вам больше по душе. И если у вас написаны объемные проекты с применением традиционных веб-форм, возможно, стоит продолжать с ними работать. Тем более, что ASP.NET Web Forms еще не умирает и также продолжает развиваться.

### MSSQL

Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

# Структура приложения

Основная работа происходит благодаря двум сервисам, которые предоставляют доступ к функциональности приложения.

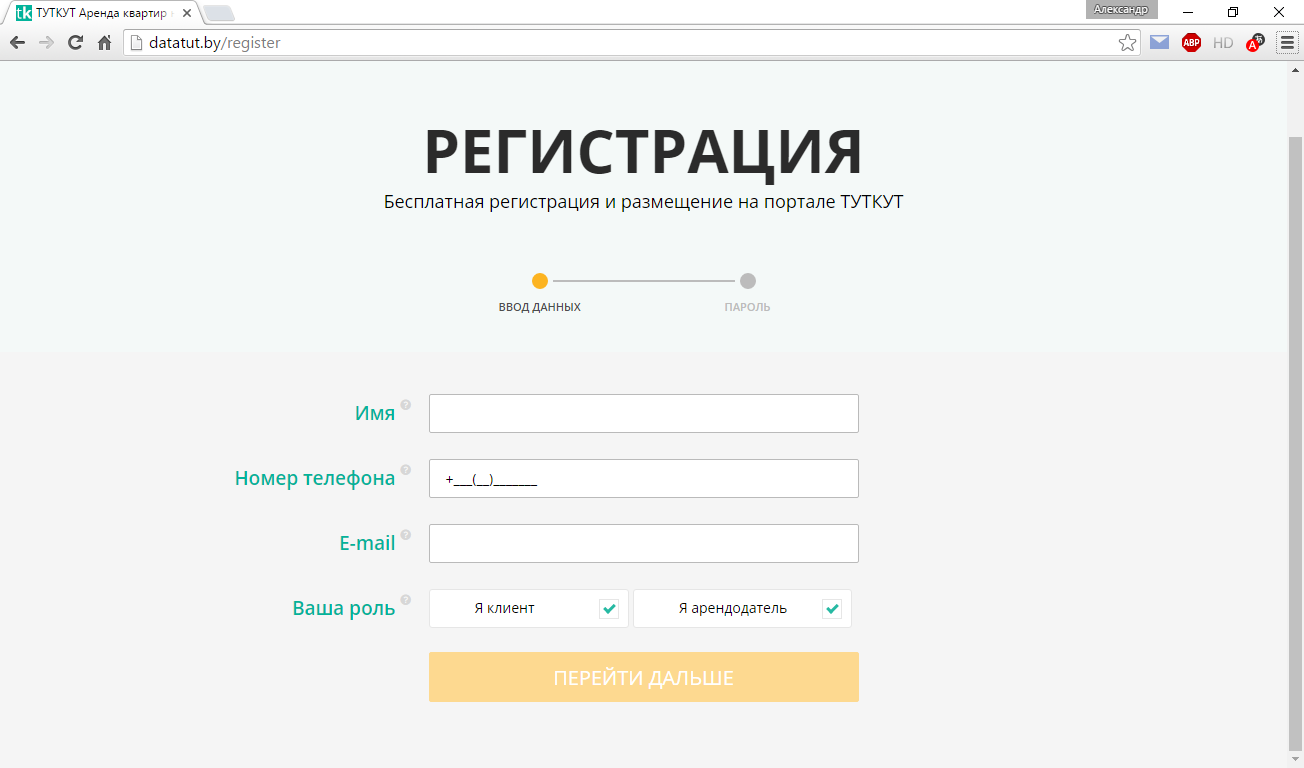
AouthService – позволяет производить манипуляции для входа, выхода пользователя из системы, так же любая операция происходит только после проверки на авторизацию.

BusinessService – воспроизводит всю бизнес-логику проекта связанную с манипуляциями с квартирами.

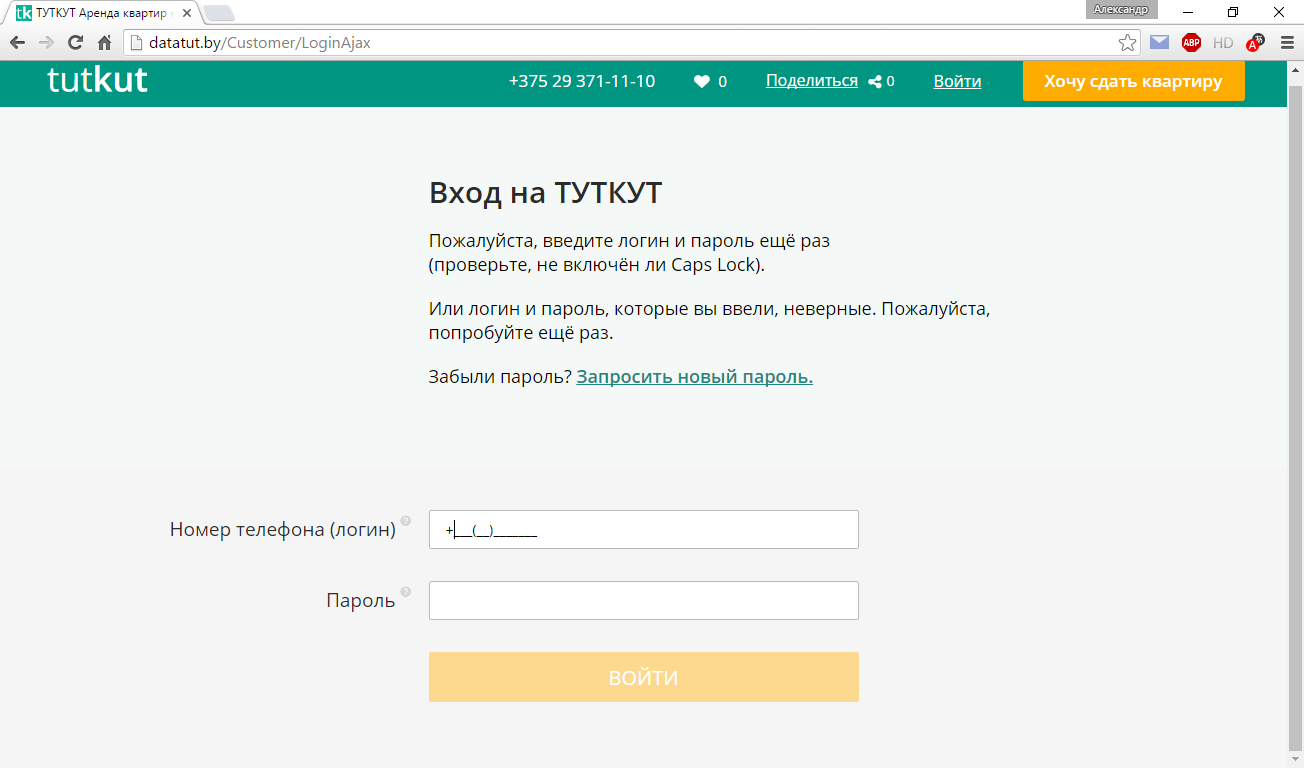
ServiceLocator – предоставляет доступ к сервисам.

Сами сервисы работают благодаря классу Repository, там реализована вся функциональность доступа к данным.

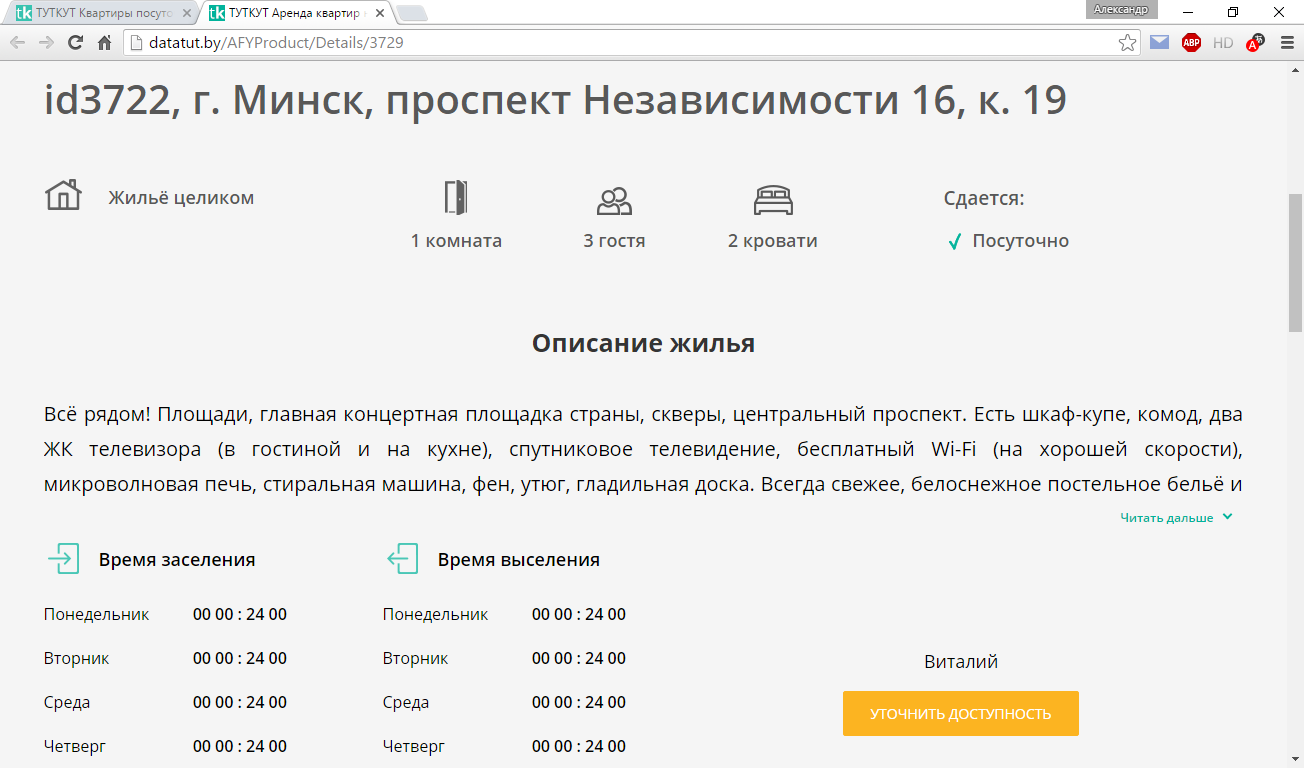
# Тестирование



Регистрация



Вход на сайт



Страница товара

# Описание применения

Данный курсовой проект можно применим как площадка, на которой люди смогут оставлять объявления и оформлять аренду квартир. Для работы приложения необходима установленная база данных MS SQL, а также .NET Framework.

# Заключение

Был разработан сервис для сдачи квартир на сутки, который позволит многим пользователям упростить процесс выбора и аренды квартиры в независимости от места проживания и особенностей. Квартиру можно выбрать по многим параметрам благодаря широкому списку характеристик для каждой квартиры в системе.

За время написания проекта была изучена информация о типах, структуре и принципах работы процессов и задач.

Были проанализированы аналогичные, из которых были выбраны те функции и логика работы, которые автор хотела бы видеть в своем программном продукте.

Разработанное программное средство представляет собой законченный программный продукт, готовый к использованию. Но при желании программу можно доработать: улучшить производительность, расширить функциональность приложения, изменить графическое оформление и др.

# Список использованных источников

1. Нейгел К. И др. C# 5.0 и платформа .NET 4.5 для профессионалов, М.: Диалектика, 2013
2. Рендольф Н., Гарднер Д., Минутилло М., Андерсон К. Microsoft Visual Studio 2010 для профессионалов, М.: Диалектика, 2011
3. Шилдт Г. C# 4.0: полное руководство, М.: ООО «И.Д. Вильямс», 2011