Содержание

[Введение 3](#__RefHeading__556_1027853445)

[1 Описание предметной области 4](#__RefHeading__558_1027853445)

[1.1 Обзор литературы 4](#__RefHeading__589_1698352192)

[1.2 Постановка задачи 4](#__RefHeading___Toc31277_289182750)

[1.3 Обоснование актуальности задачи 5](#__RefHeading___Toc2267_646782902)

[1.4 Обоснование используемых технологий 5](#__RefHeading___Toc2267_646782902)

[2 Проектирование архитектуры программно-информационной системы 6](#__RefHeading__1746_2075059964)

* 1. [Установление требований 6](#__RefHeading__1890_2075059964)

[2.1.1 Основные замечания по проекту 6](#__RefHeading__1890_2075059964)

[2.1.2 Системные сервисы 7](#__RefHeading__1890_2075059964)

[2.1.3 Системные ограничения 8](#__RefHeading__1890_2075059964)

[2.1.4 Проектные вопросы 8](#__RefHeading__1890_2075059964)

2.1.5 [Глоссарий и список используемых документов 9](#__RefHeading__1890_2075059964)

2.1.6  [Раздел предварительных замечаний 9](#__RefHeading__1890_2075059964)

[2.2 Составление спецификации требований 10](#__RefHeading__872_1456780723)

[3 Реализация программно-информационной системы 14](#__RefHeading__1892_2075059964)

[4 Результаты 17](#__RefHeading___Toc25676_289182750)

[Заключение 18](#__RefHeading__1083_1027853445)

[Список используемой литературы 19](#__RefHeading___Toc26650_289182750)

[Приложение 1 ― Код программных модулей 20](#__RefHeading__1453_2075059964)

Введение

Сегодня в мировом сообществе происходят глобальные изменения, вызванные проникновением во все сферы жизни информационных технологий. Существует огромное количество магазинов, которые завоевали огромной рынок в «офлайн» среде, но не имеют достаточной поддержки со стороны информационных технологий.

Целью данной работы является разработка архитектуры программно-информационной системы «каталог товаров магазинов и частных объявлений». Главной задачей проекта является автоматизация процесса управления продажами, как для обычных людей через «частные объявления», так и для магазинов. "Управление продажами" - это комплексное, многоплановое понятие, к которому, однако, еще не сформировалось общепринятого подхода. Некоторые специалисты рассматривают его как вопрос управления, и прежде всего людьми, которые занимаются продажами (включая подбор персонала, его мотивацию, обучение и т.д.). Другие считают, что управление продажами - это в первую очередь управление каналами сбыта. Я считаю, что довольно много внимания должно уделяться автоматизации процесса взаимодействия с клиентом.

В частности, в этой работе будет представлен модуль «Аутентификация и рассылка сообщений».

Необходимо создать прототип веб-приложения, реализующего идентификацию, авторизацию, обратную связь, возможность подписки пользователей на обновления сайта и рассылку сообщений администратором сайта. Для идентификации пользователей необходимо предоставить возможность входа через социальные сети.

Результатом разработки должно стать приложение, позволяющее регистрировать пользователей, предоставлять им различные возможности в зависимости от их роли. Также должны быть реализованы следующие возможности для пользователей: оставить отзыв о функционировании сайта или задать интересующий вопрос, подписаться на обновления сайта. Для администратора – рассылка сообщений подписавшимся на обновления пользователям.

Полный функционал системы описан далее в данной работе. Разработанная система должна корректно работать в большинстве современных веб-браузеров.

Описание предметной области

* 1. Обзор литературы

Идентификация в информационных системах — процедура, в результате выполнения которой для субъекта идентификации выявляется его идентификатор, однозначно идентифицирующий этого субъекта в информационной системе. Для выполнения процедуры идентификации в информационной системе субъекту предварительно должен быть назначен соответствующий идентификатор (т.е. проведена регистрация субъекта в информационной системе).

Процедура идентификации напрямую связана с [аутентификацией](https://ru.wikipedia.org/wiki/%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F): субъект проходит процедуру аутентификации, и если аутентификация успешна, то информационная система на основе [факторов аутентификации](https://ru.wikipedia.org/wiki/%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F) определяет идентификатор субъекта. При этом достоверность идентификации полностью определяется уровнем достоверности выполненной процедуры аутентификации.

В современном мире велика популярность различных социальных сервисов и социальных сетей - google+, twitter, facebook и т.д. Это привело разработчиков к мысли использовать в качестве логина на сайт учетную запись в данных сервисах и социальных сетях. До MVC5 это сложно было сделать. Но в MVC5 появились соответствующие механизмы, которые позволяют предельно упростить аутентификацию с помощью социальных сетей.

Выход ASP.NET MVC 5 ознаменовался переходом на новую систему авторизации под названием ASP.NET Identity. Разработчики фреймворка настоятельно рекомендуют переходить на новую систему, называя основными ее преимуществами возможность внедрения в абсолютно любой проект (ASP.NET MVC, Web Forms, Web Pages, Web API и SignalR), простую социальную интеграцию, работу на OWIN, установку и обновление посредством NuGet и другие.

В ряде случаев нам надо будет зарегистрировать свое приложение в социальных сервисах, чтобы воспользоваться возможностью подобной авторизации (как например в Facebook и vkontakte). А некоторые сервисы как Goggle и Yohoo, позволяют использовать возможность аутентификации на основе социальных сервисов без всякой регистрации.

При написании данной работы были изучены различные источники, включающие: техническую и учебно-методическую литературу, различные интернет-ресурсы. Основными источниками, раскрывающими теоретические основы механизма для идентификации пользователей, являлись работы «Начало работы с ASP.NET Identity» http://habrahabr.ru/post/227351/, «Авторизация и аутентификация в MVC5» <http://metanit.com/sharp/mvc5/12.1.php>, а также статья на тему «OWIN и Katana» <http://metanit.com/sharp/mvc5/11.1.php> и «OWIN и Katana: первый взгляд» <http://habrahabr.ru/post/202018/>.

Для разработки моделей базы данных был использован интернет-ресурс, посвященный технологии Entity Framework: <http://www.asp.net/entity-framework>.

При разработке приложения были использованы материалы из книги «OAuth 2.0 Identity and Access Management Patterns», автор Мартин Спрасовский.

Так как сайт создается с целью привлечения потенциальных клиентов или покупателей, то ресурс ориентирован в первую очередь на удобство пользователя. Разместить на страницах сайта всю информацию, которая может заинтересовать посетителя, невозможно и в этом нет необходимости. Главное, чтобы все данные, касающиеся предлагаемых товаров или услуг, были в распоряжении владельца или администратора ресурса. При этом нужно предоставить пользователю возможность быстро задать интересующий вопрос, не прикладывая для этого излишних усилий. Именно для таких целей используется форма обратной связи на сайте. Большинство данных форм работают по стандартной схеме: клиенту предлагается написать не только свой вопрос, но и имя, а также электронный адрес; это сообщение поступает на электронный ящик владельца или администратора ресурса, где он может ознакомиться с сутью вопроса; ответ отправляется непосредственно на e-mail посетителя, который он указал в форме. По такой схеме и было реализована функция «Обратная связь».

Давайте рассмотрим такую ситуацию. Обычный пользователь интернета, через поиск или любым другим путем, первый раз попадает на сайт, где получает исчерпывающую информацию, по давно интересовавшему его вопросу, либо делает вывод, что сайт удовлетворяет его интересам. Посетитель настолько остался доволен сайтом, что добавил его в закладки или записал куда-нибудь адрес сайта. Возникает вопрос вернётся ли он на сайт? Я думаю, что если ничего не напомнит ему об этом сайте, то шанс что он вернётся будет не огромным. А теперь допустим, что посетителю на е-mail пришло письмо с сообщением, что на интересующим его ресурсе была реализована новая функциональность. Исследования показывают, что на сайтах, где есть форма подписки, количество возвратов больше, примерно на 30%, чем на ресурсах, где такой подписки нет. Имея данную функциональность также можно определить на сколько пользователям интересен на данный момент сайт и на сколько важной была новая функциональность.

Основными источниками, используемыми для построения front-end части были интернет-ресурсы, посвященные фреймворкам bootstrap <http://getbootstrap.com/> и jquery <https://jquery.com/>.

* 1. Постановка задачи

Задачей этого курсового проекта является проектирование и разработка веб-приложения, реализующего следующий функционал: идентификация и авторизация пользователей, обратная связь, возможность подписки пользователей на обновления сайта и рассылка сообщений администратором сайта.

Пользователи сайта должны быть уверены, что их данные надёжно защищены и никто другой не должен иметь доступ к этим данным. Также нужно разграничить права доступа пользователей к ресурсам сайта. Для решения этой задачи подходит реализация механизма Identity 2.0, он обладаем рядом следующих преимуществ: возможность внедрения в любой проект, реализована работа с ролями через использование стандартных атрибутов MVC [Authorize], пароли хранятся в хешированном виде, есть возможность использования данных от социальных сетей и общеизвестных сайтов, таких как twitter.com, vk.com и т.д. для входа на сайт.

* 1. Обоснование актуальности задачи

Довольно важной задачей при разработке веб-сайтов и веб-приложений есть ограничение доступа к некоторым разделам сайта, например, к панели администратора. Без разделения функциональных возможностей между пользователями невозможно достигнуть высокого качества и безопасности. Сайт будет обладать некоторыми функциональными возможностями (подтверждение/отклонение заявок от пользователей на создание магазина в каталоге и т.д.), которые не должны быть доступны обычным пользователям, а должны быть доступны только администраторам. Хеширование паролей защитит систему в случае попадания базы данных злоумышленникам.

Форма обратной связи требуется для отсылки сообщений напрямую с сайта, посетители скорее согласятся написать сообщение через данную форму, чем через электронный ящик, к тому же если оставлять свой электронный ящик он может попасть в руки «спамеров». Ко всему вышесказанному - обратная связь обычно ведёт к улучшению качества продукта.

1.4 Обоснование выбранных технологий

Используемые технологии:

* entity framework. Выбран потому что предоставляет возможность работы с базами данных через объектно-ориентированный код C#. Этот подход предоставляет ряд существенных преимуществ: нам не нужно беспокоиться о коде доступа к данным, нам не нужно знать деталей работы СУБД SQL Server и синтаксиса языка запросов T-SQL, вместо этого мы работаем с таблицами базы данных как с классами C#, с полями этих таблиц - как со свойствами классов, а синтаксис SQL-запросов, который в ADO.NET раньше нужно было вставлять в код C# в виде команд, заменен на более удобный подход с LINQ. Entity Framework берет на себя обязанности по преобразованию кода C# в SQL-инструкции;
* шаблон проектирования MVC. Предполагает разделение данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: Модель, Представление и Контроллер – таким образом, что модификация каждого компонента может осуществляться независимо;
* bootstrap. Самый популярный UI framework для разработки адаптивных и мобильных web-проектов. Bootstrap использует самые современные технологии CSS и HTML. С помощью этого фреймворка проект легко и эффективно масштабируется с одной базой кода, от телефонов и планшетов до настольных компьютеров. На сайте разработчиков bootstrap можно найти обширную и прекрасную документации с сотнями живых примеров, фрагментов кода.
* ASP.NET identity 2.0. По сравнению с предыдущими версиями возможна двухфакторная аутентификация, подтверждение аккаунта через электронную почту, восстановление пароля, использование социальных сетей для входа на сайт.

2. Проектирование архитектуры программно-информационной системы

* 1. Установление требований
     1. Основные замечания по проекту

Цели и рамки проекта

Целью данного проекта является разработка информационной системы для размещения информации о товарах магазинов и частных объявлений. Через реализованную систему магазины приобретут потенциальных покупателей, а пользователям будет предоставлен каталог товаров и объявлений. Название системы - ИС «Каталог товаров магазинов и частных объявлений».

Деловой контекст

Мы хотим создать площадку, которая будет максимально полезна магазинам и покупателям. Как показывает официальная статистика, основные нарушения в интернет магазинах связаны с отсутствием полной информации о продавце и о самом товаре. Нередко предоставляется недостоверная информация о продавце и о товаре. Веб-портал даст возможность для каждого продавца размещать информацию о существующих товарах, что поможет увеличить продажи, количество звонков в офис, привлечь новых посетителей на сайт и сформировать постоянную аудиторию, также каждый пользователь сможет размещать частные объявления. Вход пользователей на сайт будет возможен через социальные сети и пользователям будет предоставлена возможность оставить отзыв или задать интересующий вопрос.

Участники проекта

Заказчик:

Разработчики: Малахов В.В., Лавицкий Е.Д., Мисников А.Г.

Идеи в отношении решений

Программа должна быть реализована в виде web-приложения для любых операционных систем. Web-приложение будет разворачиваться на виртуальном хостинге, изначально на Azure. Разработка будет проходить по методологии Scrum. Для управления жизненный циклом приложения будет использоваться Visual Studio Team Services, где будут планироваться работы, размещаться задачи и баги.

Обзор документа

В разделе «Системные сервисы» описывается требования к разрабатываемой информационной системе. В разделе «Системные ограничения» определяются критерии и ограничения разрабатываемой системы при обслуживании и дальнейшей поддержке.

В разделе «Проектные вопросы» освещаются прочие проектные вопросы.

* + 1. Системные сервисы

Рамки системы можно моделировать с помощью диаграммы контекста

представленной на рисунке 1.



Рисунок 1 - Контекстная диаграмма «GoodShops App»

Функциональные требования

ИС должна обеспечивать следующие функциональные возможности:

* работа с пользователями
  + авторизация и аутентификация;
  + идентификация
  + вход через социальные сети
* управления заявками на добавление магазинов;
* обратная связь

Требования к данным

ИС должна хранить свои данные в специализированной базе, пароли должны храниться в хешированном виде. Резервная копия базы данных должна делаться 1 раз в день.

* + 1. Системные ограничения

Требования к интерфейсу

ИС должна иметь интуитивно-понятный интерфейс, не требующий долгого изучения и разработанный для удобства выполнения сложных операций. Интерфейс приложения должен иметь частичный альтернативный способ взаимодействия (через горячие клавиши) с элементами управления. Также приложение должно иметь адаптивный дизайн элементов управления.

Требования к производительности

Приложение должно корректно работать на системах с поддержкой IIS7 и старше, которая имеет не меньше 8Гб оперативной памяти и жёсткий диск объёмом не меньше 250Гб.

Требования к безопасности

ИС не должна раскрывать личных данных пользователей.

Эксплуатационные требования

ИС должна функционировать на ОС Windows 7, ОС Windows 8, ОС Windows 8.1, ОС Windows 10. ИС требует 8гб оперативной памяти, процессор не ниже Intel i5 с частотой не ниже 2.2Ггц и 2 физических ядра.

Политические и юридические требования

Нет.

Другие ограничения

Не выявлены.

* + 1. Проектные вопросы

Список открытых вопросов:

* интерфейс приложения;
* сторонние библиотеки для использования;
* как привлечь клиентов

Предварительный план-график

1.09.2015 – 1.10.2015 – Анализ и установление требований к ИС

1.10.2015 – 1.11.2015 – Спецификация требований к ИС

1.11.2015 – 1.12.2015 – Кодирование ИС

1.12.2015 – 31.12.2015 – Тестовая эксплуатация ИС

11.01.2016 – 13.01.2016 – Ввод в эксплуатацию

Предварительный бюджет

Предварительный бюджет отсутствует.

* + 1. Глоссарий и список используемых документов

Глоссарий

ПО – программной обеспечение

ИС – информационная система

ОС – операционная система

MVC – Model View Controller (шаблон проектирования разделяющий бизнес-логику, данные и их представление)

Деловые документы и формы

ГОСТ 2,105.95 — Оформление текстовых документов

* + 1. Раздел предварительных замечаний

Сравнение программы с аналогами.

Основные отличия решения от аналогов:

* использование ajax-запросов, это позволяет веб приложению выполнять все свои функции без полной перезагрузки страницы
* применение адаптивной вёрстки
  1. Составление спецификации требований

Диаграмма иерархии классов, взаимодействующих на сервере представлена на рисунке 2.

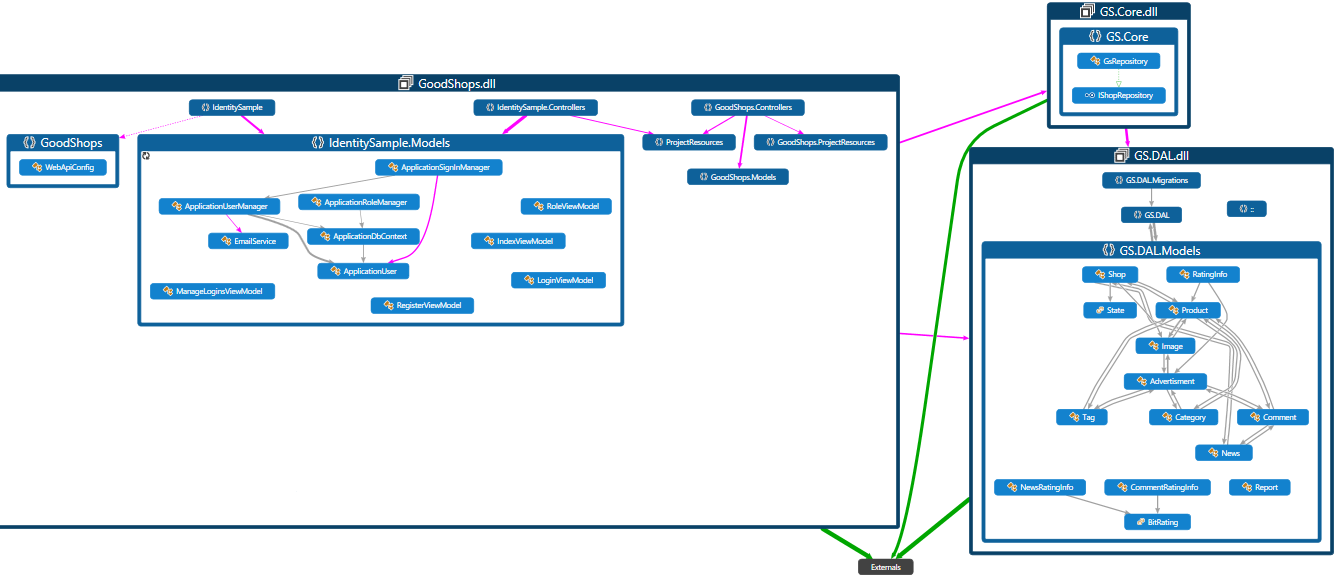


Рисунок 2 - Диаграмма иерархии классов

Диаграмма прецедентов, выявленных во время процесса проектирования представлена на рисунке 3.

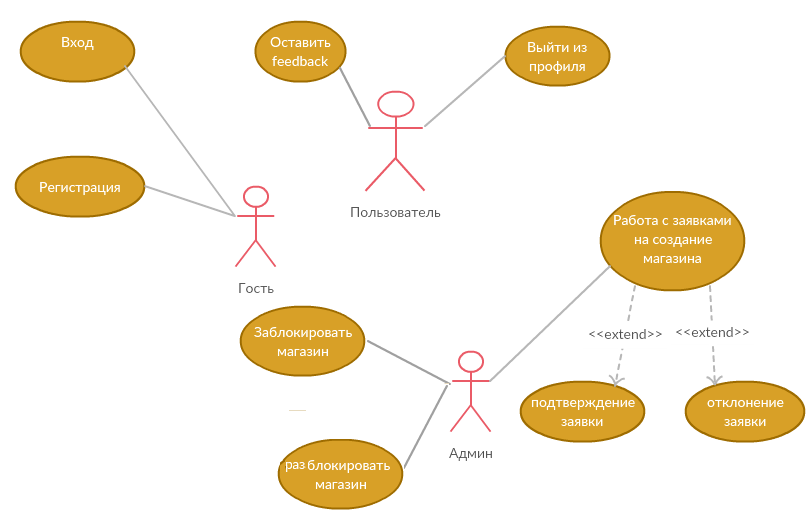


Рисунок 3 - Диаграмма прецедентов «Роли, заявки, обратная связь»

ТУТ СТРАНИЦ 8 НА ТАБЛИЦЫ

Диаграммы последовательностей представлены на рисунках 4 и 5.

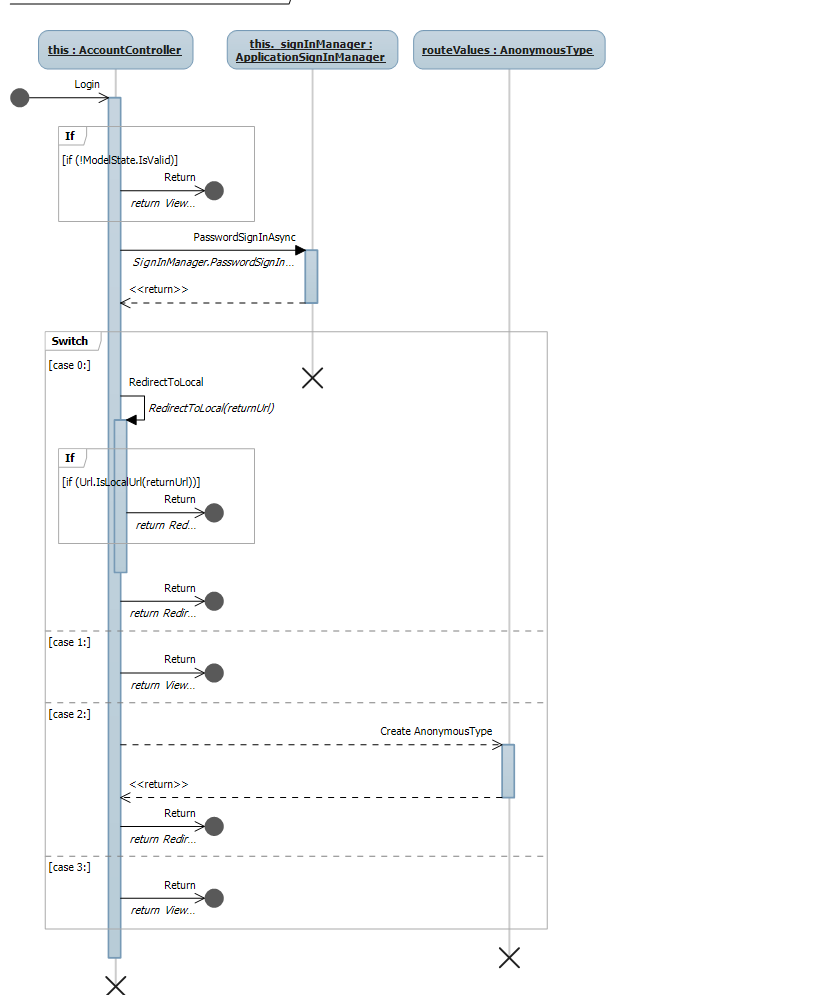


Рисунок 4 - Диаграмма последовательностей для входа пользователей

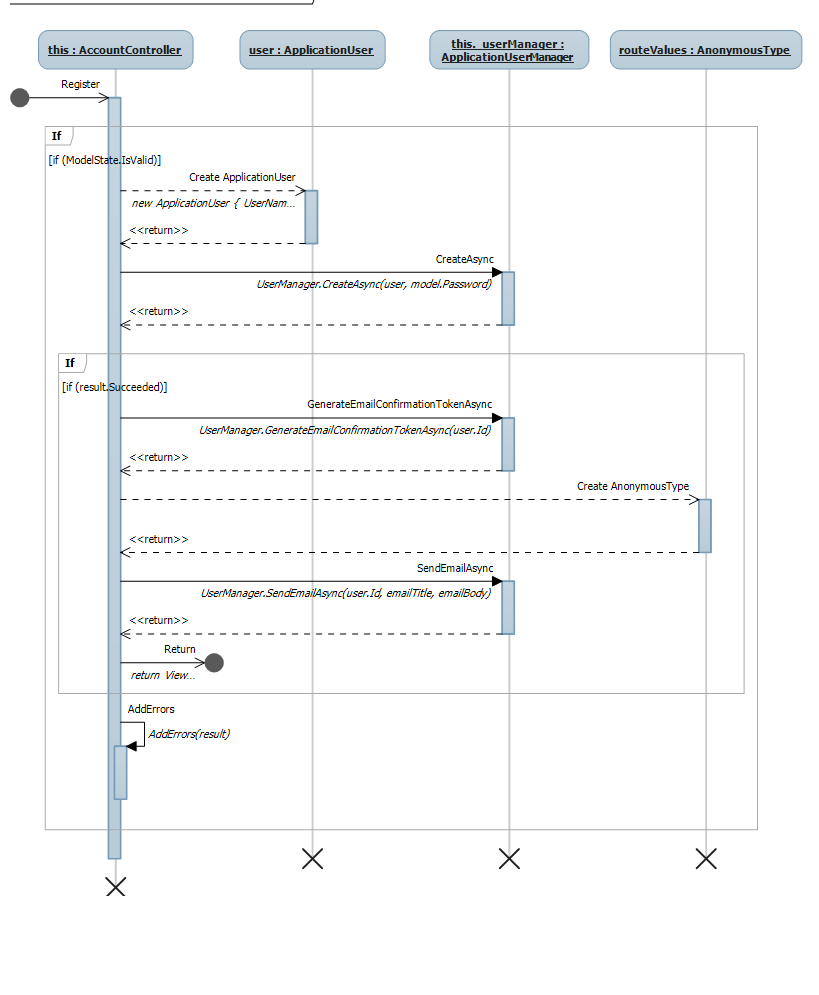


Рисунок 5 - Диаграмма последовательностей для регистрации

пользователей

2.3. Проектирование структуры слоя доступа к данным

Диаграмма классов модели выявленных во время проектирования представлена на рисунке 6.

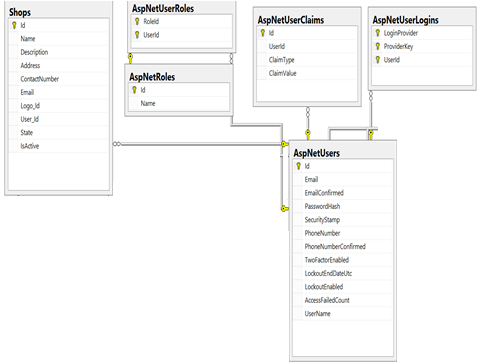


Рисунок 6 - Диаграмма классов модели данных

1. Реализация программно-информационной системы

3.1. Описание бизнес логики

3.2. Пример использования программно-информационной системы

Шапка страницы состоит из меню навигации, которое включаете в себя 6 элементов для пользователей, не являющимися администраторами: название сайта, главная, о нас, обратная связь и элементы для идентификации пользователя. Для пользователя, который является администратором на меню навигации присутствует ещё один элемент – управление магазинами.

Меню навигации для обычного пользователя представлено на рисунке 7.



Рисунок 7 – Меню навигации не для администратора

Меню навигации для администратора представлено на рисунке 8.

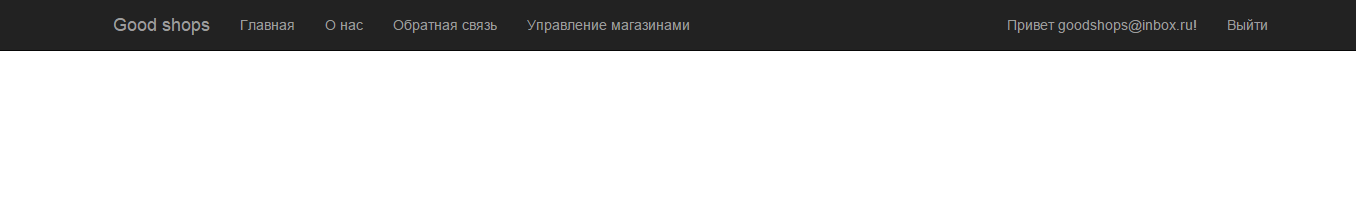


Рисунок 8 - Меню навигации для администратора

Меню навигации может состоять из разного количества элементов, в зависимости от роли пользователя.

При нажатии на кнопку «Главная» пользователь будет перенаправлен на начальную страницу сайта.

При нажатии на кнопку «О нас», будет открыта страница, изображённая на рисунке 9.

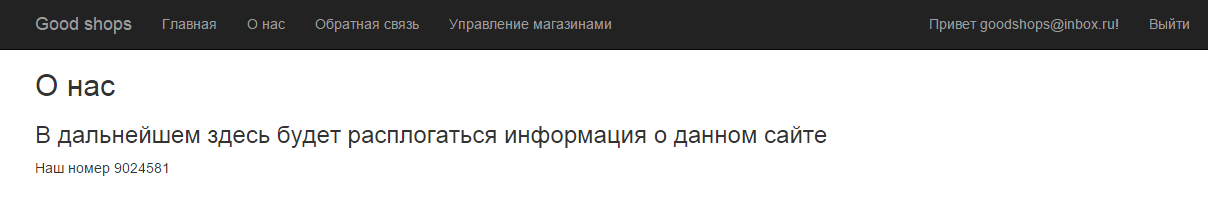


Рисунок 9 - Страница «О нас»

При нажатии на кнопку «Обратная связь», будет открыта страница, изображённая на рисунке 10.

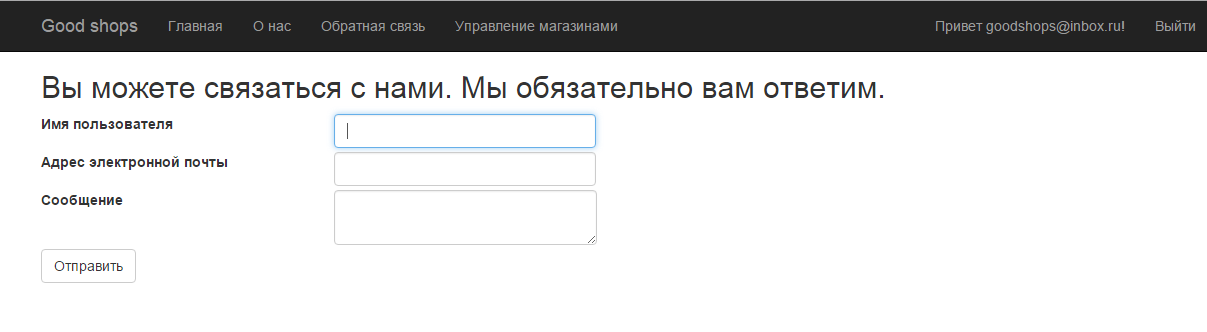


Рисунок 10 – Страница «Обратная связь»

При нажатии на кнопку «Управление магазинами», будет открыта страница, изображённая на рисунке 11.

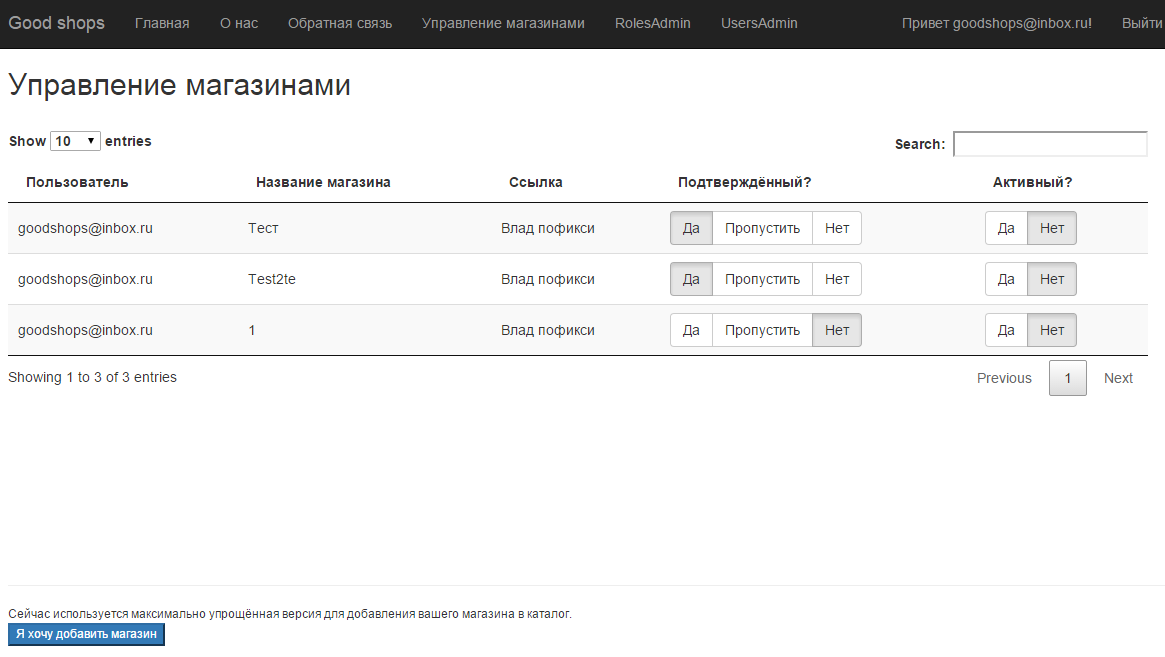


Рисунок 11 – Страница «Управление магазинами»

При нажатии на кнопку «Регистрация», будет открыта страница, изображённая на рисунке 12.

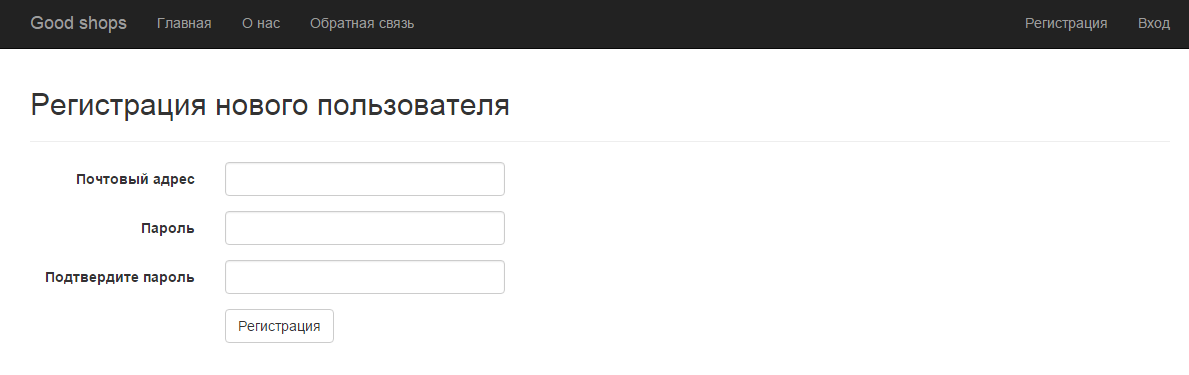


Рисунок 12 – Страница «Регистрация»

При нажатии на кнопку «Вход», будет открыта страница, изображённая на рисунке 13.

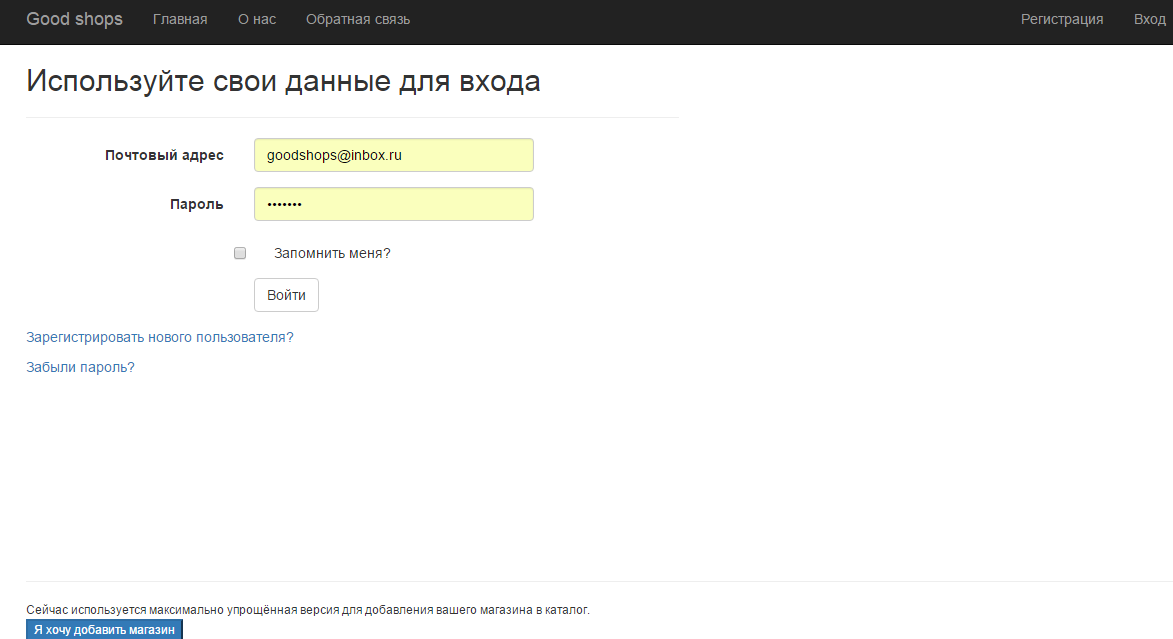


Рисунок 13 – Страница «Вход»

4 Результаты

Выполнен основной этап разработки данной ИС. Установлены новые требования к её дальнейшему развитию. Удалось добиться стабильной работы приложения при разных типах входных данных. Получены необходимые знания для оптимизации и «облегчения» частично реализованной системы. Разработана первичная техническая документация в виде данной пояснительной записки. Проведено функциональное тестирование программных модулей.

Заключение

Разработан первый модуль информационной системы. Продумана базовая архитектура приложения и реализовано 35% модуля «Обратной связи» и 80% модуля «Идентификации пользователей» намеченного в ходе планирования функционала. Реализовано централизованное хранилище и логика по хранению динамичных и статичных данных ИС.

Разработанная ИС «Goodshops App» обеспечивает следующие функциональные возможности:

* авторизация и аутентификация пользователей;
* идентификация пользователей;
* обратная связь;
* управления заявками магазинов

В ходе выполнения данной курсовой работы были приобретены необходимые навыки при реализации крупных ИС с использованием шаблона проектирование MVC, Identity 2.0, ajax-запросов, а также стилей bootstrap.

Список используемой литературы

1. Биллинг, В. А. Основы программирования на C#. / В. А. Биллинг. – М: Бином, 2012 – 488 с.
2. Васильев, А. Н. C#. Объектно-ориентированное программирование. Учебный курс. / А. Н. Васильев. — СПб. : Питер. 2012.—320 с: ил.
3. Спрасовский, Мартин. OAuth 2.0 Identity and Access Management Patterns. / Мартин Спрасовский. — Packt Publishing, 2013. - 109 с.
4. Подбельский, В. В. Язык С#. Решение задач / В. В. Подбельский. — М: Инфра-М, 2014. – 296 с.
5. Троелсен, Э. Язык программирования С# 4.0 и платформа .NET 4.0 : пер. с англ. / Э. Троелсен. – М. : Вильямс, 2011. – 1392 с.
6. Шилдт, Г. C# 4.0 : полное руководство : пер. с англ. / Г. Шилдт. – М. : Вильямс, 2011 – 1056 с.
7. Visual C# 2010. Полный курс. / Уотсон К [и др.]. – М. : Вильямс, 2011. – 955 с.