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 посетителей 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](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 большим. 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 такими функциональными возможностями, как подтверждение/отклонение заявок от пользователей на создание магазина в каталоге, рассылка сообщений пользователям, подписанным на обновления сайта, 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.

Подписка на обновления сайта необходима для уведомления подписавшихся пользователей о новой функциональности. Исследования показывают, что на сайтах, где есть форма подписки, количество возвратов больше, примерно на 30%, чем на ресурсах, где такой подписки нет.

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 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;
* асинхронные запросы ajax. Основные достоинства: возможность создания удобного Web–интерфейса, активное взаимодействие с пользователем, частичная перезагрузка страницы, вместо полной, удобство использования;
* отправка сообщения через SmtpClient. Позволяет приложениям отправлять электронную почту с помощью протокола SMTP (Simple Mail Transfer Protocol). Подключение к серверу SMTP, установленное текущим экземпляром класса SmtpClient, может использоваться повторно, если приложению потребуется отправить несколько сообщений тому же серверу SMTP. Имеется возможность для асинхронной отправки сообщений, а также возможность для прикрепления файлов к сообщению

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, задать интересующий вопрос, подписаться на обновления.

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 будут 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;
* блокировка и разблокировка магазинов;
* обратная связь;
* подписка на обновления сайта;
* рассылка сообщений из аккаунта администратора пользователям, подписанным на обновления сайта

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

2.2 Составление спецификации требований

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

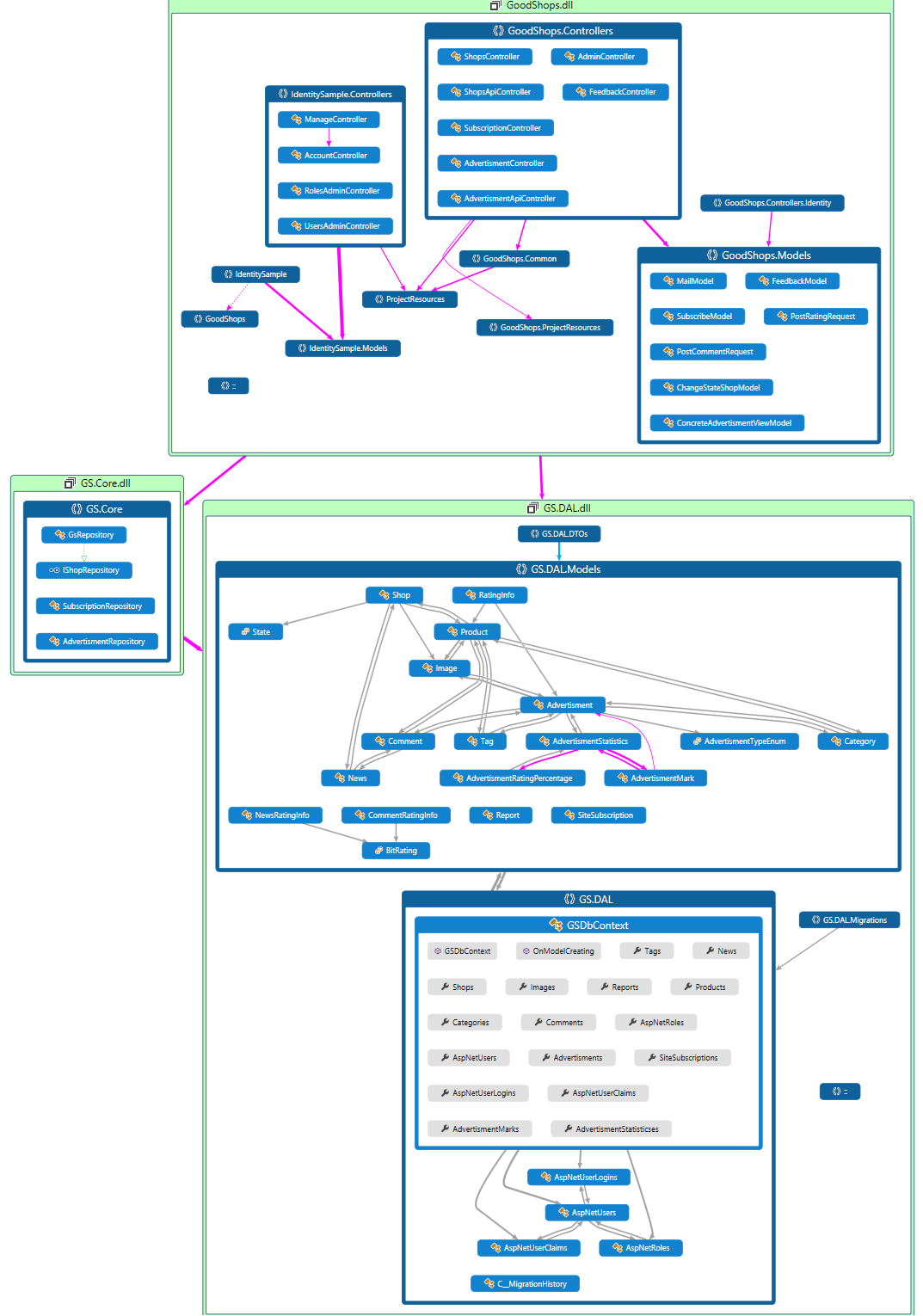


Рисунок 2 – Диаграмма иерархии классов

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

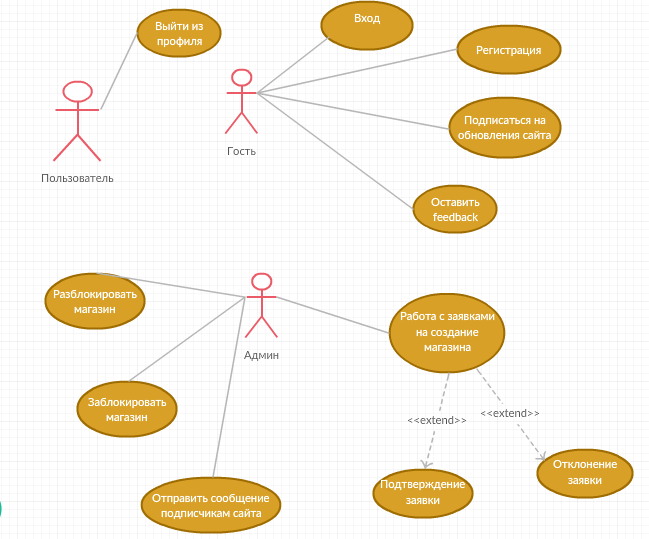


Рисунок 3 – Диаграмма вариантов использования системы

Таблица 1.1 – Определение варианта использования «Вход»

|  |  |
| --- | --- |
| Идентификатор | UC-01 |
| Наименование | Вход |
| Первичное действующее лицо | Гость |
| Другие действующие лица | Нет |
| Описание | Пользователь, который имеет аккаунт в системе может войти на сайт, где система выполнит процедуру аутентификации. Возможен вход через социальные сети «в контакте» и другие. |
| Предварительные условия | Есть соединение с базой данных. Пользователь находится на странице входа. |
| Выходные условия | Если логин и пароль соответствуют существующим в системе, пользователь получает соответствующие ему права. |
| Нормальный поток развития | UC-01.0 Вход   1. Перейти на страницу входа. 2. Заполнить данные. 3. Нажать на кнопку «Войти». |
| Альтернативный поток развития | UC-01.1 Вход   1. Перейти на страницу входа. 2. Нажать на кнопку «Vkontakte». 3. Подтвердить вход на открывшейся странице. |
| Исключения | UC-01.Е.1 Ошибка заполнения данных:  Если пользователь не заполнит все обязательные поля, то он увидит ошибки валидации данных.  UC-01.Е.2 Ошибка соединения с базой данных:  Наличие такой ошибки не позволит зафиксировать изменения в базе данных. |
| Включение | Нет |
| Частота использования | Используется редко. |
| Ссылки | Нет |
| Предположения | Нет |
| Замечания и вопросы | Нет |

Таблица 1.2 – Определение варианта использования «Регистрация»

|  |  |
| --- | --- |
| Идентификатор | UC-02 |
| Наименование | Регистрация посетителя сайта |
| Первичное действующее лицо | Гость |
| Другие действующие лица | Авторизованный пользователь |
| Описание | Посетитель регистрируется в системе |
| Предварительные условия | Есть соединение с базой данных. Пользователь находится на странице регистрации. |
| Выходные условия | Пользователь получает аккаунт в системе, если введённые им данные успешно прошли валидацию. |
| Нормальный поток развития | UC-02.0 Регистрация   1. Перейти на страницу регистрации. 2. Заполнить данные. 3. Нажать на кнопку «Регистрация». |
| Альтернативный поток развития | Нет |
| Исключения | UC–02.Е.1 Ошибка заполнения данных:  Если пользователь не заполнит все обязательные поля, то он увидит ошибки валидации данных.  UC–02.Е.2 Ошибка соединения с базой данных:  Наличие такой ошибки не позволит зафиксировать изменения в базе данных. |
| Включение | Нет |
| Частота использования | В большинстве случаев используется 1 раз. |
| Ссылки | Нет |
| Предположения | Нет |
| Замечания и вопросы | Нет |

Таблица 1.3 – Определение варианта использования «Выйти из профиля»

|  |  |
| --- | --- |
| Идентификатор | UC-03 |
| Наименование | Выйти из профиля |
| Первичное действующее лицо | Авторизованный пользователь |
| Другие действующие лица | Нет |
| Описание | Авторизованный пользователь выходит из системы |
| Предварительные условия | Авторизованный пользователь. Есть соединение с базой данных. |
| Выходные условия | Пользователю доступны только действия для незарегистрированных пользователей. |
| Нормальный поток развития | UC-03.0 Выйти из профиля   1. Нажать на кнопку «Выйти» в главном меню |
| Альтернативный поток развития | Нет |
| Исключения | UC-03.Е.1 Ошибка соединения с базой данных:  Наличие такой ошибки не позволит зафиксировать изменения в базе данных. |
| Включение | Нет |
| Частота использования | Используется редко. |
| Ссылки | Нет |
| Предположения | Нет |
| Замечания и вопросы | Нет |

Таблица 1.4 – Определение варианта использования «Оставить feedback»

|  |  |
| --- | --- |
| Идентификатор | UC-04 |
| Наименование | Оставить feedback |
| Первичное действующее лицо | Гость |
| Другие действующие лица | Авторизованный пользователь |
| Описание | Пользователь оставляет отзыв |
| Предварительные условия | Есть соединение с базой данных. Пользователь находится на странице «Обратная связь». |
| Выходные условия | На почту администратору приходит письмо. |
| Нормальный поток развития | UC-04.0 Оставить feedback   1. Нажать на элемент «Обратная связь» в главном меню. 2. Заполнить данные. 3. Нажать на кнопку «Отправить». |
| Альтернативный поток развития | Нет |
| Исключения | UC-04.Е.1 Ошибка соединения с базой данных:  Наличие такой ошибки не позволит зафиксировать изменения в базе данных.  UC-04.Е.2 Ошибка заполнения данных:  Если пользователь не заполнит все обязательные поля, то он увидит ошибки валидации данных.  UC-04.E.3 Недопустимые настройки stmp:  Наличие такой ошибки не позволит отправить сообщение с текущими настройками smtp. У пользователя откроется страница, на который будет написано, что отправка не удалась. |
| Включение | Нет |
| Частота использования | Используется часто. |
| Ссылки | Нет |
| Предположения | Нет |
| Замечания и вопросы | Нет |

Таблица 1.5 – Определение варианта использования «Подписаться на обновления сайта»

|  |  |
| --- | --- |
| Идентификатор | UC-05 |
| Наименование | Подписка на обновления сайта |
| Первичное действующее лицо | Пользователь |
| Другие действующие лица | Нет |
| Описание | Пользователь подписывается на обновления сайта и будет получать сообщения об обновлениях, которые будет отправлять администратор. |
| Предварительные условия | Есть соединение с базой данных. Пользователь находится на главной странице. |
| Выходные условия | Пользователь увидит сообщение что он подписался на обновления сайта и ему на почту будут отправляться сообщения об обновлениях от администратора сайта. |
| Нормальный поток развития | UC-05.0 Подписаться на обновления сайта   1. Перейти на главную страницу. 2. Нажать на блок с надписью «подписаться». 3. Заполнить данные. 4. Нажать на кнопку «Подписаться». |
| Альтернативный поток развития | Нет |
| Исключения | UC-05.Е.1 Ошибка соединения с базой данных:  Наличие такой ошибки не позволит зафиксировать изменения в базе данных.  UC-05.Е.2 Ошибка заполнения данных:  Если пользователь не заполнит все нужные поля, то он увидит ошибки валидации данных. |
| Включение | Нет |
| Частота использования | Обычно используется 1 раз. |
| Ссылки | Нет |
| Предположения | Нет |
| Замечания и вопросы | Нет |

Таблица 1.6 – Определение варианта использования «Отправить сообщение подписчиками сайта»

|  |  |
| --- | --- |
| Идентификатор | UC-06 |
| Наименование | Отправить сообщение подписчикам сайта |
| Первичное действующее лицо | Администратор |
| Другие действующие лица | Нет |
| Описание | Администратор можно уведомлять пользователей, которые подписаны на обновления сайта. Подписчикам приходит сообщение на почту. |
| Предварительные условия | Авторизированный пользователь является администратором. Есть соединение с базой данных. Администратор находится на странице «Рассылка». |
| Выходные условия | Все пользователи, подписанные на обновления сайта, получили сообщение на почту, которое написал администратор. |
| Нормальный поток развития | UC-06.0 Отправить сообщение подписчиками сайта   1. Перейти на страницу «Рассылка» 2. Заполнить заголовок и текст сообщения. 3. Нажать на кнопку «Отправить» |
| Альтернативный поток развития | Нет |
| Исключения | UC-06.Е.1 Ошибка заполнения данных:  Если пользователь не заполнит заголовок или текст сообщения, то он увидит ошибки валидации  UC-06.Е.2 Ошибка соединения с базой данных:  Наличие такой ошибки не позволит зафиксировать изменения в базе данных. |
| Включение | Нет |
| Частота использования | Часто |
| Ссылки | Нет |
| Предположения | Нет |
| Замечания и вопросы | Нет |

Таблица 1.7 – Определение варианта использования «Заблокировать магазин»

|  |  |
| --- | --- |
| Идентификатор | UC–06 |
| Наименование | Заблокировать магазин |
| Первичное действующее лицо | Администратор |
| Другие действующие лица | Нет |
| Описание | Администратор может заблокировать магазин, если он видит нарушения со стороны магазина. |
| Предварительные условия | Авторизированный пользователь является администратором. Есть соединение с базой данных. Администратор находится на странице «Управление магазинами». |
| Выходные условия | Магазин заблокирован и не виден в списке магазинов для пользователей. |
| Нормальный поток развития | UC-07.0 Заблокировать магазин   1. Перейти на страницу «Управление магазинами». 2. В таблице со списком магазинов в строке магазина в столбце «Активный?» нажать на кнопку «Нет». 3. Подтвердить решение в всплывающем окне. |
| Альтернативный поток развития | Нет |
| Исключения | UC-07.Е.1 Ошибка соединения с базой данных:  Наличие такой ошибки не позволит зафиксировать изменения в базе данных. |
| Включение | Нет |
| Частота использования | Редко |
| Ссылки | Нет |
| Предположения | Нет |
| Замечания и вопросы | Нет |

Таблица 1.8 – Определение варианта использования «Разблокировать магазин»

|  |  |
| --- | --- |
| Идентификатор | UC-08 |
| Наименование | Разблокировать магазин |
| Первичное действующее лицо | Администратор |
| Другие действующие лица | Нет |
| Описание | Администратор может разблокировать магазин, если все нарушения магазина были исправлены. |
| Предварительные условия | Авторизированный пользователь является администратором. Есть соединение с базой данных. Администратор находится на странице «Управление магазинами». |
| Выходные условия | Магазин разблокирован и доступен пользователям в общем списке магазинов. |
| Нормальный поток развития | UC-08.0 Разблокировать магазин   1. Перейти на страницу «Управление магазинами» 2. В таблице со списком магазинов в строке магазина в столбце «Активный?» нажать на кнопку «Да». 3. Подтвердить решение в всплывающем окне. |
| Альтернативный поток развития | Нет |
| Исключения | UC-08.Е.1 Ошибка соединения с базой данных:  Наличие такой ошибки не позволит зафиксировать изменения в базе данных. |
| Включение | Нет |
| Частота использования | Редко |
| Ссылки | Нет |
| Предположения | Нет |
| Замечания и вопросы | Нет |

На рисунке 4 представлена диаграмма последовательности варианта использования «Вход».

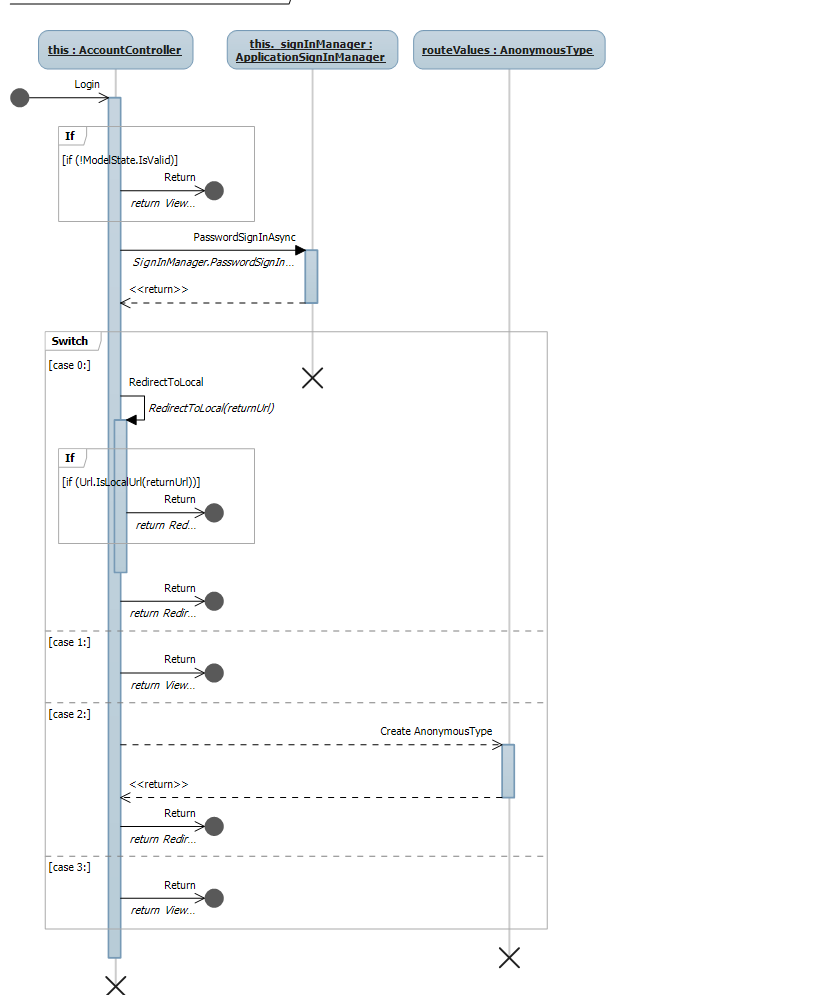


Рисунок 4 – Диаграмма последовательности варианта использования «Вход»

На рисунке 5 представлена диаграмма последовательности варианта использования «Регистрация».

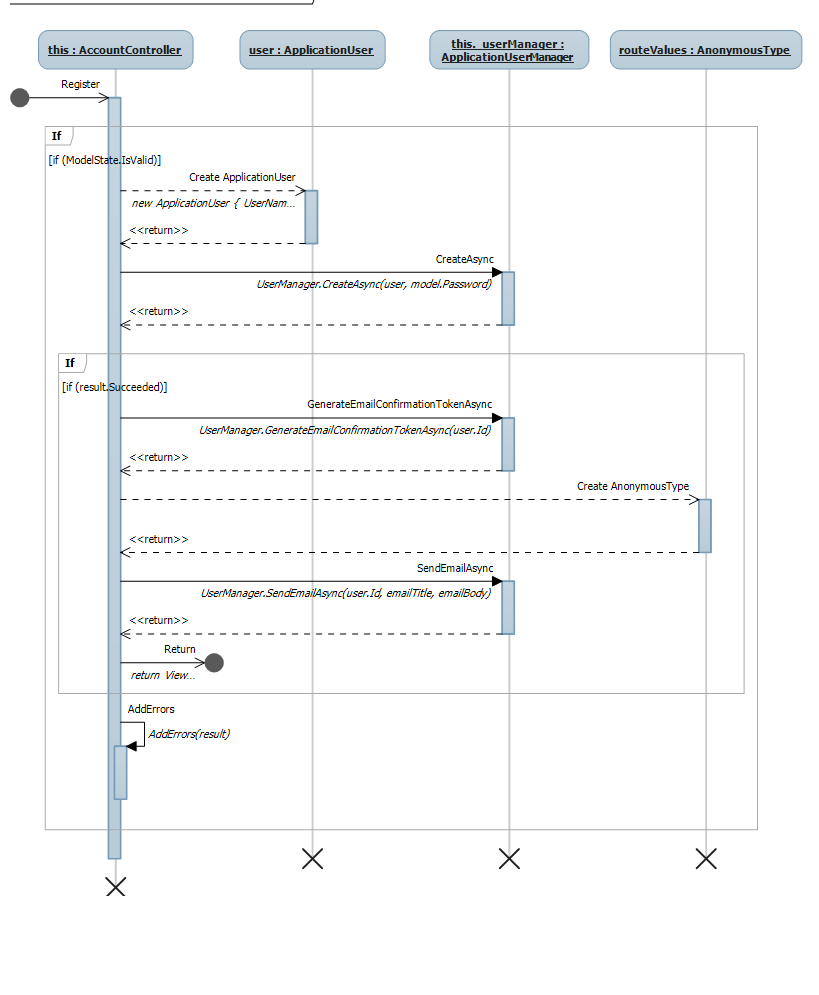


Рисунок 5 – Диаграмма последовательностей варианта использования «Регистрация»

На рисунке 6 представлена диаграмма последовательности варианта использования «Выйти из профиля».

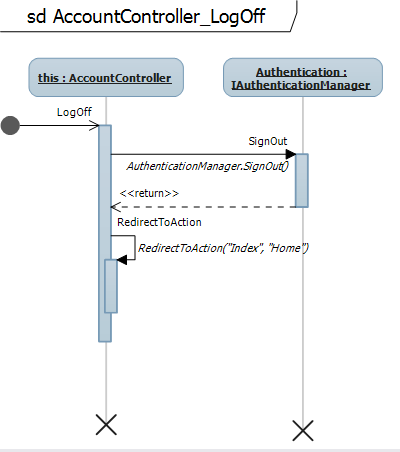


Рисунок 6 – Диаграмма последовательностей варианта использования «Выйти из профиля»

На рисунке 7 представлена диаграмма последовательности варианта использования «Оставить feedback».

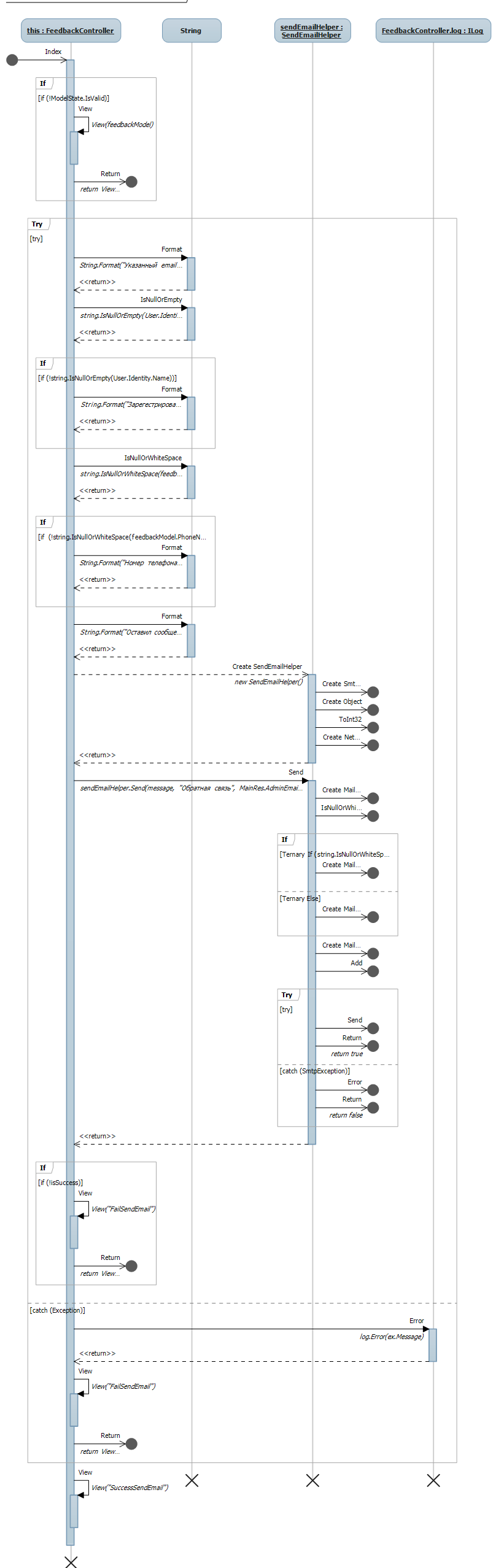


Рисунок 7 – Диаграмма последовательностей варианта использования «Оставить feedback»

На рисунке 8 представлена диаграмма последовательности варианта использования «Подписаться на обновления сайта».



Рисунок 8 – Диаграмма последовательностей варианта использования «Подписаться на обновления сайта»

На рисунке 9 представлена диаграмма последовательности варианта использования «Отправить сообщение подписчиками сайта».

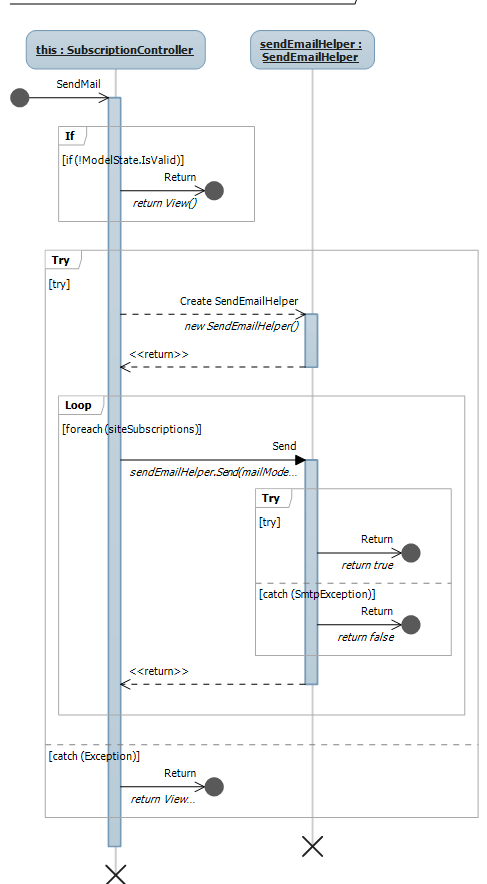


Рисунок 9 – Диаграмма последовательностей варианта использования «Отправить сообщение подписчиками сайта»

На рисунке 10 представлена диаграмма последовательности вариантов использования «Заблокировать/разблокировать магазин».

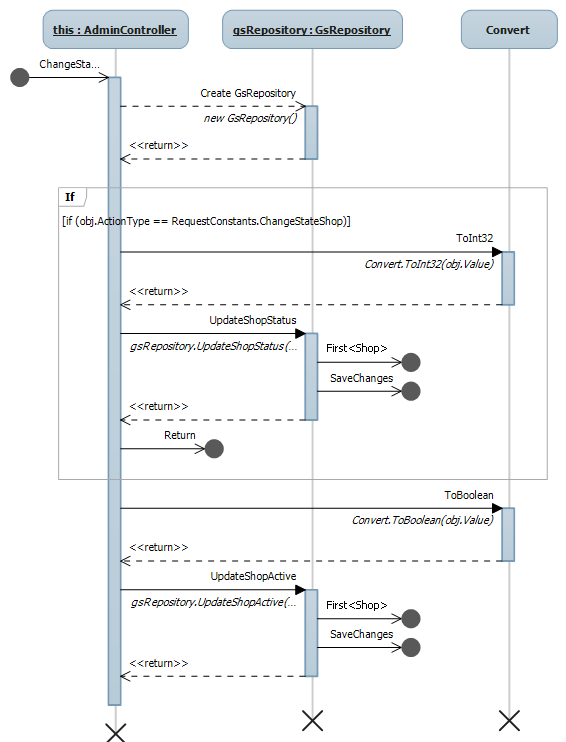


Рисунок 10 – Диаграмма последовательностей вариантов использования «Заблокировать/разблокировать магазин»

2.3 Проектирование структуры слоя доступа к данным

Диаграмма классов модели выявленных во время проектирования представлена на рисунке 11.

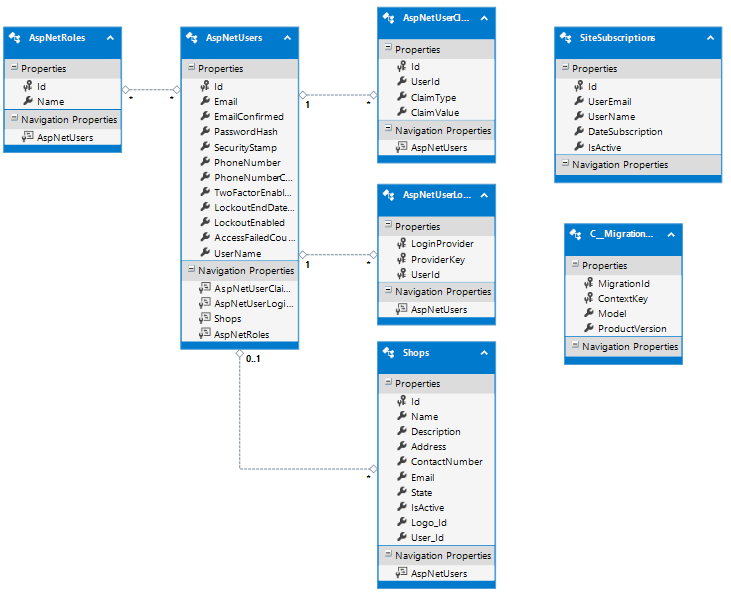


Рисунок 11 – Диаграмма классов модели данных

На рисунке 12 представлена структура таблицы «AspNetUsers».

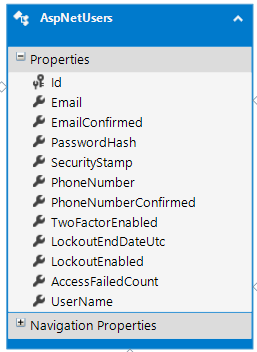


Рисунок 12 – Структура таблицы «AspNetUsers»

Сущность «AspNetUsers» представляет конкретного пользователя, она содержит следующие поля:

– Id (уникальный номер);

– Email (почтовый адрес);

– EmailConfirmed (подтверждена ли почта);

– PasswordHash (пароль в хешированном виде);

– SecurityStamp (штамп безопасности, если пароль пользователя изменится – изменится и штамп, что приведёт к отклонению «cookies», которые использовались ранее)

– PhoneNumber (номер телефона);

– PhoneNumberConfirmed (подтверждён ли номер телефона);

– TwoFactorEnabled (включена ли двухфакторная аутентификация);

– LockoutEndDateUTC (дата и время до какого аккаунт будет заблокирован);

– LockoutEnabled (включена ли блокировка аккаунта);

– AccessFailedCount (допустимое количество раз ввода неправильного пароля подряд, после чего аккаунт будет заблокирован, при условии, что LockoutEnabled = true);

– UserName (имя пользователя)

На рисунке 13 представлена структура таблицы «AspNetUserLogins».

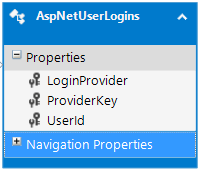


Рисунок 13 – Структура таблицы «AspNetUserLogins»

Сущность «AspNetUserLogins» представляет пользователя, который логинится с помощью внешних сервисов, она содержит следующие поля:

– LoginProvider (логин провайдера);

– ProviderKey (уникальный номер провайдера)

– UserId (внешний ключ к таблице AspNetUsers, данное значение содержит id пользователя, связь один–ко–многим, так как пользователь может логинится через различных провайдеров

На рисунке 14 представлена структура таблицы «AspNetUserClaims».

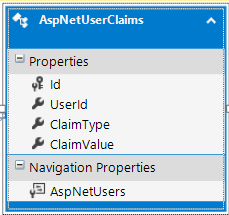


Рисунок 14 – Структура таблицы «AspNetUserClaims»

Сущность «AspNetUserClaims» хранит «claim». «Claim» представляет иную модель авторизации по сравнению с ролями. Грубо говоря, «claim» содержит некоторую информацию о пользователе, например, адрес электронный почты, логин, возраст и т.д. И эта информация позволяет идентифицировать пользователя и наделить его соответствующими правами доступа. Данная сущность содержит следующие поля:

– Id (уникальный номер);

– UserId (внешний ключ к таблице AspNetUsers, данное значение содержит id пользователя);

– ClymeType (тип «clime»)

На рисунке 15 представлена структура таблицы «AspNetRoles».

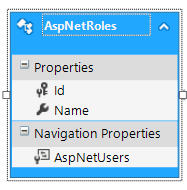


Рисунок 15 – Структура таблицы «AspNetRoles»

Сущность «AspNetRoles» представляет конкретную роль, она содержит следующие поля:

– Id (уникальный номер);

– Name (имя роли)

На рисунке 16 представлена структура таблицы «AspNetUserRoles».

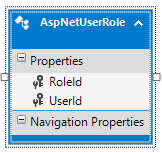


Рисунок 16 – Структура таблицы «AspNetUserRoles»

Сущность «AspNetUserRoles» представляет связь между пользователем и ролью, она содержит следующие поля:

– UserId (уникальный номер пользователя);

– RoleId (уникальный номер роли)

На рисунке 17 представлена структура таблицы «Shop».

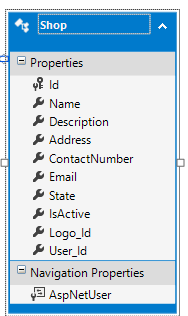


Рисунок 17 – Структура таблицы «Shop»

Сущность «Shop» представляет конкретный магазин, она содержит следующие поля:

– Id (уникальный номер);

– Name (название магазина);

– Description (краткое описание магазина);

– Address (адрес магазина);

– ContactNumber (телефон магазина);

– Email (электронная почта магазина);

– Logo\_Id (внешний ключ к таблице Images, хранящей url картинок, магазин может иметь одну главную картинку, связь один–к–одному);

– User\_Id (внешний ключ к таблице AspNetUsers, хранящий зарегистрированных пользователей, у каждого магазина есть владелец, связь один–к–одному);

– State (состояние магазина, может быть: подтвержден, не подтвержден, ожидает проверки);

– IsActive (значение, которое показывает доступен ли сейчас магазин)

На рисунке 18 представлена структура таблицы «SiteSubscriptions».

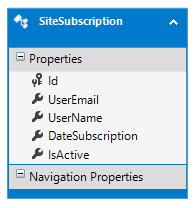


Рисунок 18 – Структура таблицы «SiteSubscriptions»

Сущность «SiteSubscriptions» представляет конкретную подписку на обновления сайта, она содержит следующие поля:

– Id (уникальный номер);

– UserEmail (электронная почта пользователя, подписавшегося на обновления);

– UserName (имя пользователя);

– DateSubscription (дата, когда пользователь подписался на обновления сайта);

– IsActive (активна ли подписка)

3 Реализация программно–информационной системы

3.1 Описание бизнес логики

Класс SendEmailHelper предоставляет функциональность для отправки сообщений.

На рисунке 19 представлена диаграмма класса «SendEmailHelper».

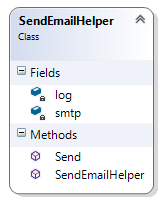


Рисунок 19 – Диаграмма класса «SendEmailHelper»

Описание полей и методов класса:

– log (поле, используемое для логирования данных);

– smtp (поле, для которого задаётся smtp сервер и которое используется для отправки сообщений);

– Send (метод для отправки сообщений)

Метод используемый для отправления ajax–запроса для подписки пользователя на обновления сайта, имеет вид:

function confirmSiteSubscribe(requestData) {

BootstrapDialog.show({

title: 'Отлично!',

message: 'Вы уверены?',

type: BootstrapDialog.TYPE\_SUCCESS,

buttons: [

{

label: 'Да',

action: function (dialog) {

dialog.close();

$.ajax('Home/SiteSubscribe', {

type: 'POST',

data: JSON.stringify(requestData),

contentType: "application/json; charset=utf–8",

success: function () {

BootstrapDialog.alert({

title: 'Успешно',

message: 'Вы были подписаны на обновления сайта!',

type: BootstrapDialog.TYPE\_SUCCESS

});

},

error: function (errMesage) {

BootstrapDialog.alert({

title: 'Внимание',

message: ' Не удалось подписаться на обновления сайта!' + errMesage,

type: BootstrapDialog.TYPE\_DANGER

});

}

});

}

},

{

label: 'Нет',

action: function (dialog) {

dialog.close();

}

}

]

});

}

Параметр requestData данного метода представляет объект, который имеет следующие свойства:

– Email (электронный адрес подписчика);

– UserName (имя пользователя)

Подписка происходит на главной странице сайта в следующем скрипте

$(document).ready(function () {

$('#btnSubsribe').click(function () {

if ($('#pnlSubsribe').css('display') == 'none')

$('#pnlSubsribe').fadeIn();

else

$('#pnlSubsribe').fadeOut();

});

$('#btnSubsribe2').click(function () {

var emailName = $('#txtEmail').val();

if (emailName) {

$('#validEmail').hide();

var requestData = {

Email: emailName,

UserName: $('#txtUserName').val(),

};

confirmSiteSubscribe(requestData);

}

else $('#validEmail').show();

});

});

Обработка ajax–запроса на сервере представлена следующим методом

public void SiteSubscribe(SubscribeModel subscribe)

{

try

{

if (!ModelState.IsValid) return;

var subscriptionRepository = new SubscriptionRepository();

subscriptionRepository.Subscribe(subscribe.Email, subscribe.UserName);

}

catch (Exception ex)

{

log.Error(ex.Message);

}

}

Метод Subscribe класса SubscriptionRepository имеет следующий вид

public void Subscribe(string email, string name)

{

if (string.IsNullOrWhiteSpace(email)) return;

if (context.SiteSubscriptions.FirstOrDefault(s => s.UserEmail == email) != null)

return;

var subscription = new SiteSubscription();

subscription.IsActive = true;

subscription.UserEmail = email;

subscription.UserName = name;

subscription.DateSubscription = DateTime.Now;

context.SiteSubscriptions.Add(subscription);

context.SaveChanges();

}

Класс SubscriptionRepository находится в проекте бизнес–логики и служит промежуточным звеном между веб-приложением и базой данных, его назначение – управление подписками пользователей на обновления сайта.

На рисунке 20 представлена диаграмма класса «SubscriptionRepository».

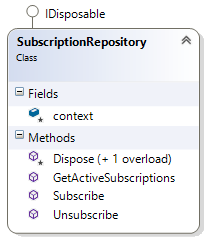


Рисунок 20 – Диаграмма класса «SubscriptionRepository»

Описание полей и методов класса:

– context (поле, используемое для получения, изменения данных в базе данных);

– GetActiveSubscriptions (метод, который получает список подписчиков, у которых активирована подписка);

– Subscribe (метод для подписки пользователя на обновления сайта);

– Unsubscribe (метод для отписки пользователя от обновлений сайта)

Для управления магазинами используется представленный ниже java–script:

function manageShop(button) {

var type = BootstrapDialog.TYPE\_WARNING;

var message = "";

if (button.name == 'isActive') {

if (button.value == 'true') {

message = "Магазин будет добавлен в каталог? Вы согласны?";

} else {

message = "Магазин не прошёл модерацию. Вы уверены?";

}

operation = "changeActiveShop";

} else {

if (button.value == '1') {

message = "Магазин будет разблокирован. Продолжить?";

} else {

message = "Магазин будет заблокирован. Вы уверены?";

}

operation = "changeStateShop";

}

var requestData = {

ActionType: operation,

ShopId: $(button).parents().eq(3).find('td').first().text().trim(),

Value: button.value

};

BootstrapDialog.show({

title: 'Подтвердите изменение информации!',

message: message,

type: type,

buttons: [

{

label: 'Да',

action: function (dialog) {

dialog.close();

$.ajax('ChangeStatusShop', {

type: 'POST',

data: JSON.stringify(requestData),

contentType: "application/json; charset=utf–8",

success: function () {

BootstrapDialog.alert({

title: 'Успешно',

message: 'Изменения были схранены!',

type: BootstrapDialog.TYPE\_SUCCESS

});

},

error: function (errMesage) {

BootstrapDialog.alert({

title: 'Внимание',

message: 'Не удалось сохранить!' + errMesage,

type: BootstrapDialog.TYPE\_DANGER

});

}

});

}

},

{

label: 'Нет',

action: function (dialog) {

dialog.close();

}

}

]

});

}

Подписка происходит на странице управления магазинами в следующем скрипте

$(document).ready(function () {

$("input[type='radio']").on('change', function () {

if (this.value == '0') {

$(this).parents().eq(3).remove();

return;

}

manageShop(this);

});

$('#tableManageShops').DataTable({

});

});

Обработка ajax–запроса на сервере представлена следующим методом

public void ChangeStatusShop(ChangeStateShopModel obj)

{

var gsRepository = new GsRepository();

if (obj.ActionType == RequestConstants.ChangeStateShop)

{

gsRepository.UpdateShopStatus(obj.ShopId, Convert.ToInt32(obj.Value));

return;

}

gsRepository.UpdateShopActive(obj.ShopId, Convert.ToBoolean(obj.Value));

}

Класс GsRepository находится в проекте бизнес–логики и служит промежуточным звеном между веб–приложением и базой данных, его назначение – получение, изменения, управление данными о магазинах.

На рисунке 21 представлена диаграмма класса «GsRepository».

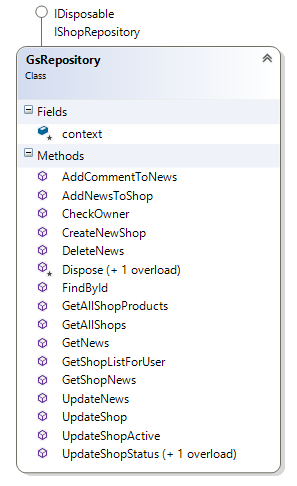


Рисунок 21 – Диаграмма класса «GsRepository»

Описание используемых полей и методов класса:

– context (поле, используемое для получения, изменения данных в базе данных);

– GetAllShops (метод, который получает список магазинов);

– UpdateShopActive (метод, который блокирует либо снимает блокировку с магазина);

– UpdateShopStatus (метод, который обновляет статус магазина на подтверждённый или не подтверждённый)

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 два элемента – управление магазинами и рассылка.

Меню навигации для обычного пользователя представлено на рисунке 22.

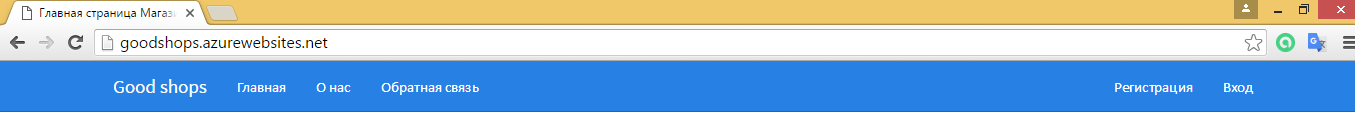


Рисунок 22 – Меню навигации для обычного пользователя

Меню навигации для администратора представлено на рисунке 23.

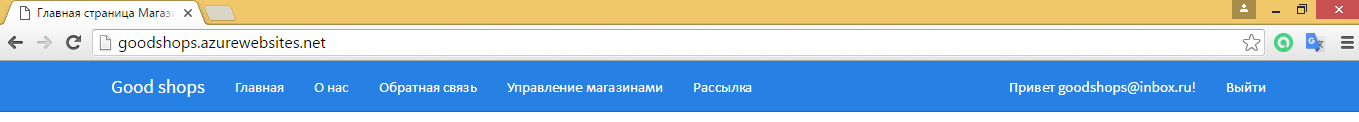


Рисунок 23 – Меню навигации для администратора

Когда посетитель впервые попадает на сайт, он находится на главной странице, она представлена на рисунке 24.

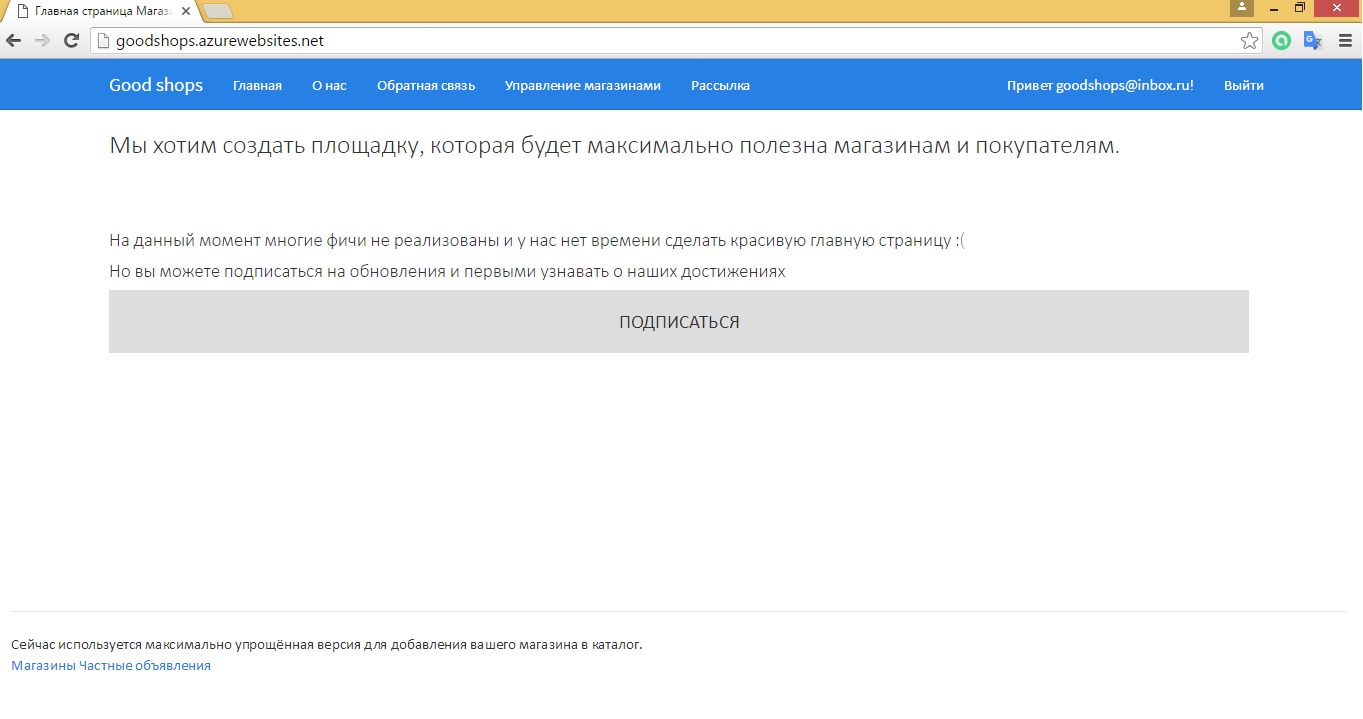


Рисунок 24 – Главная страница сайта

На главной странице пользователь может подписаться на обновления сайта. Для этого нужно нажать на серый блок с надписью «подписаться» после чего пользователь увидит форму, которую необходимо заполнить для подписки на обновления сайта, представленную на рисунке 25.

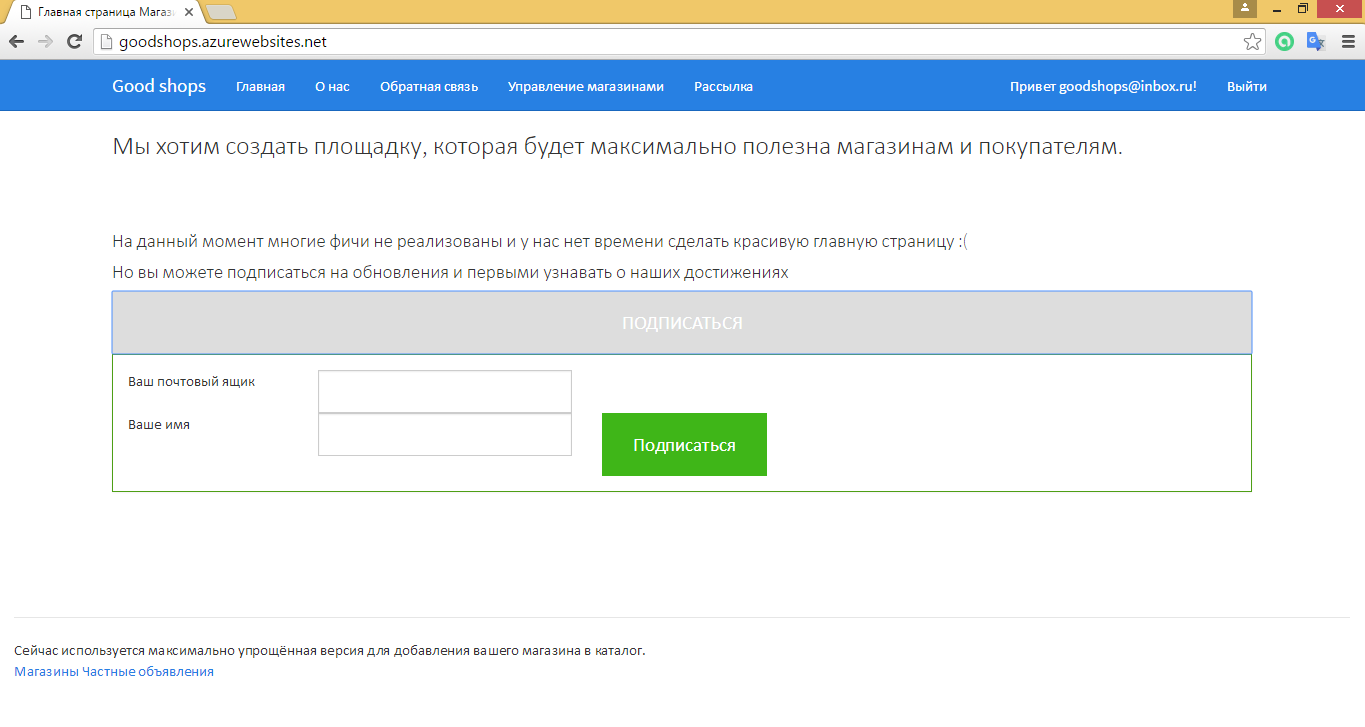


Рисунок 25 – Форма для подписки пользователя на обновления сайта

Почтовый ящик является обязательным полем для заполнения, т.к. все уведомления будут приходить на него. В случае, если пользователь не введёт его он увидит результат, представленный на рисунке 26.

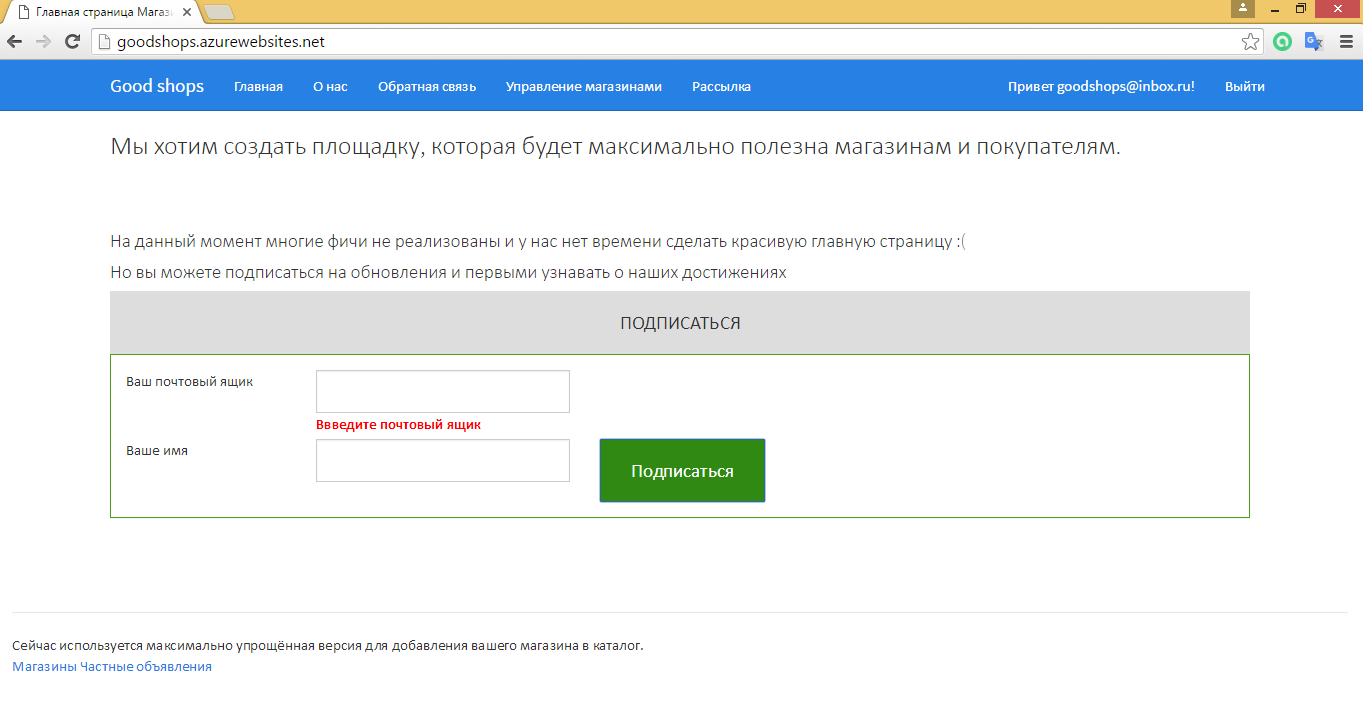


Рисунок 26 – Валидация при подписке на обновления сайта

Если данные были заполнены корректно система попросит подтверждения данного действия, результат представлен на рисунке 27.

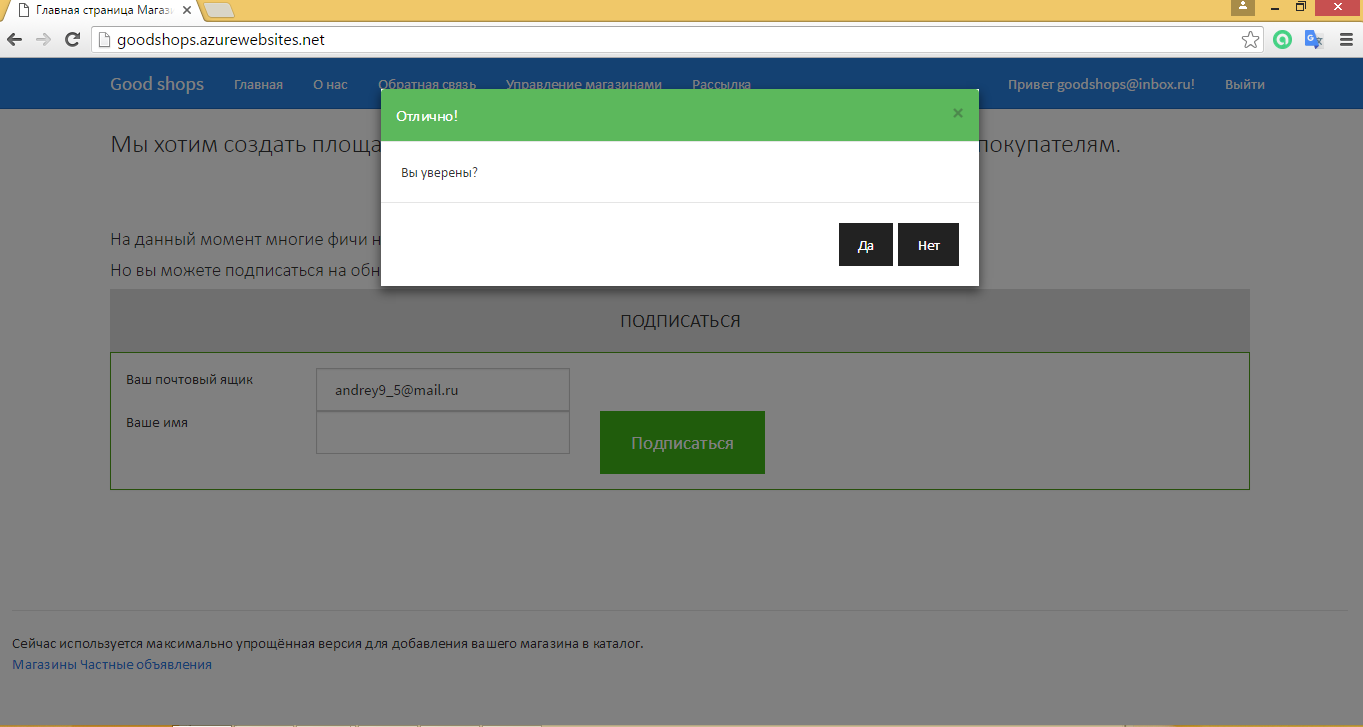


Рисунок 27 – Подписка на обновления сайта, с успешно пройденной валидацией

После подтверждения пользователем текущего действия будет отправлен запрос на сервер, если пользователь подпишется на обновления он увидит сообщение, представленное на рисунке 28.

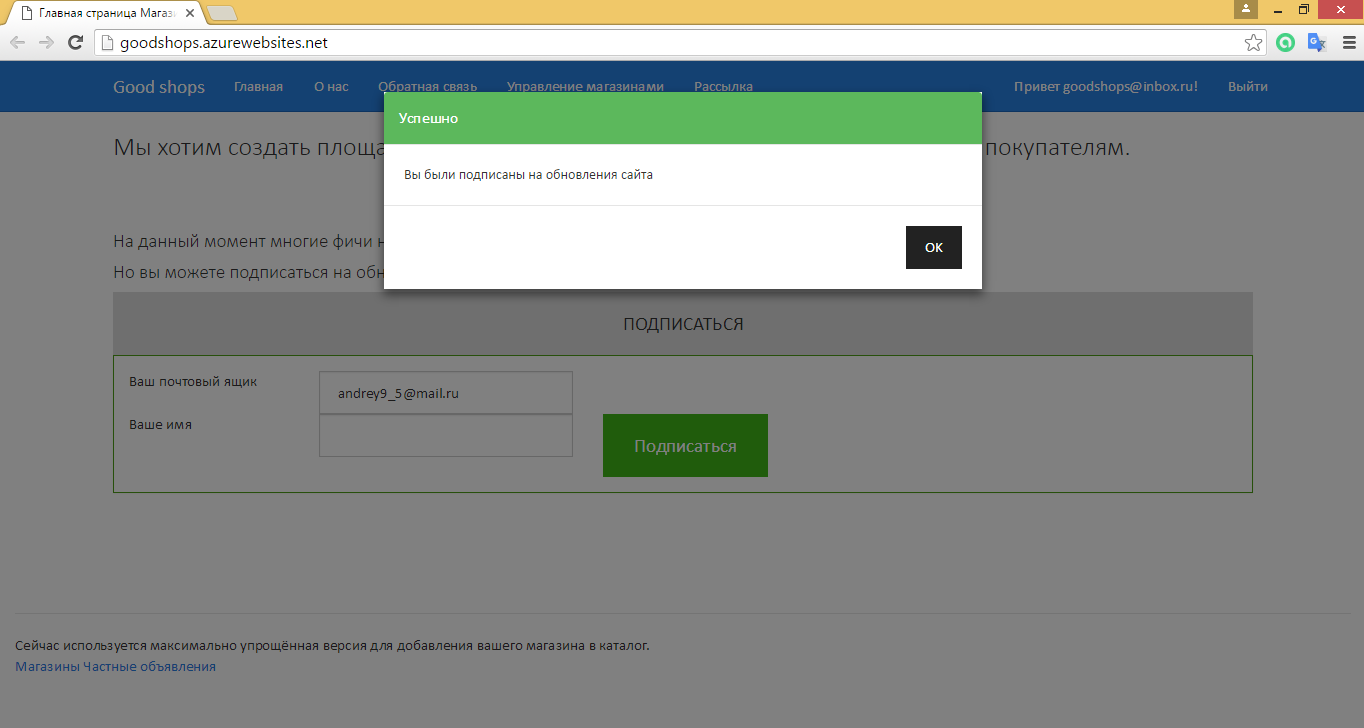


Рисунок 28 – Уведомления пользователя об успешной подписке на обновления сайта

Для того чтобы оповестить пользователей о новой функциональности на сайте, администратору необходимо открыть страницу «Рассылка», она представлена на рисунке 29.

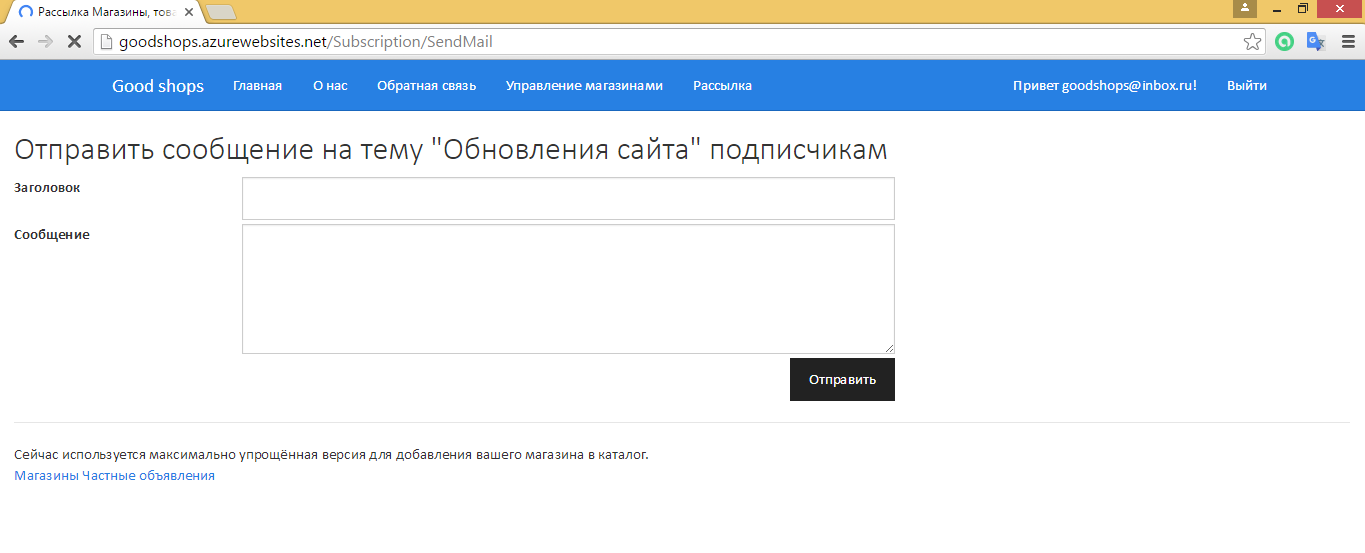


Рисунок 29 – Страница для рассылки администратором сообщений подписанным на обновления пользователям

На данной странице для отправки сообщения должны быть обязательно заполнены поля «Заголовок» и «Сообщение», иначе администратор увидит сообщения об ошибках, представленные на рисунке 30.

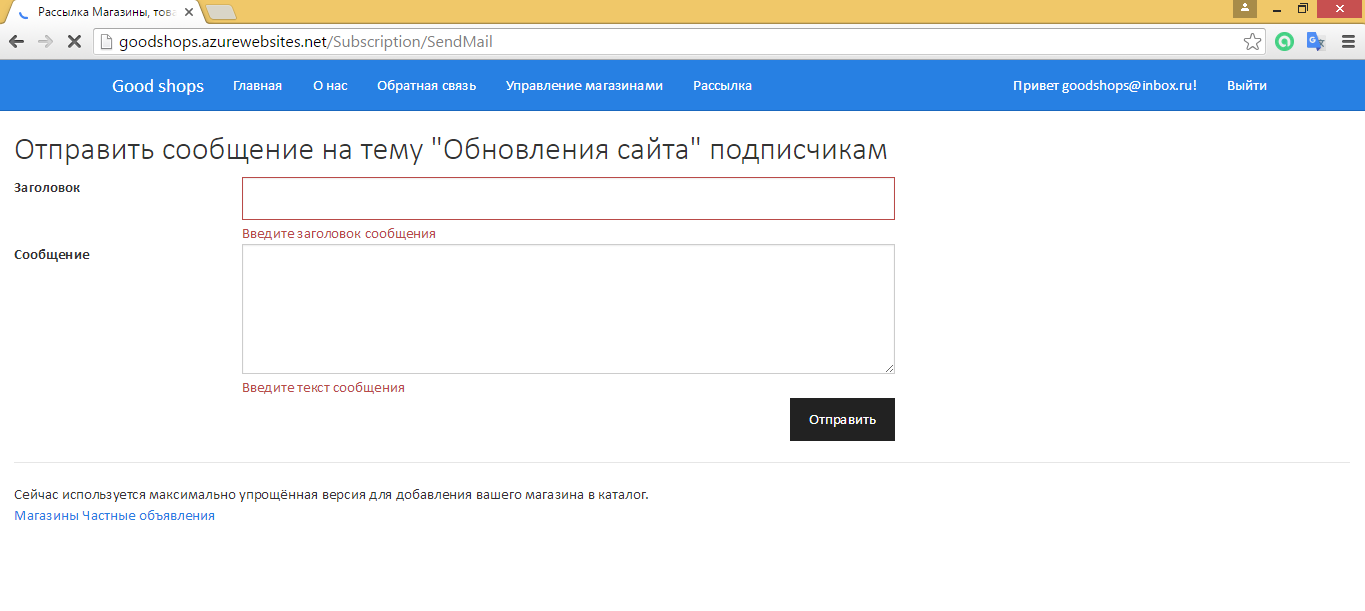


Рисунок 30 – Валидация на странице «Рассылка»

После заполнения всех данных и нажатия на кнопку отправить, администратор перенаправится на страницу, представленную на рисунке 31.

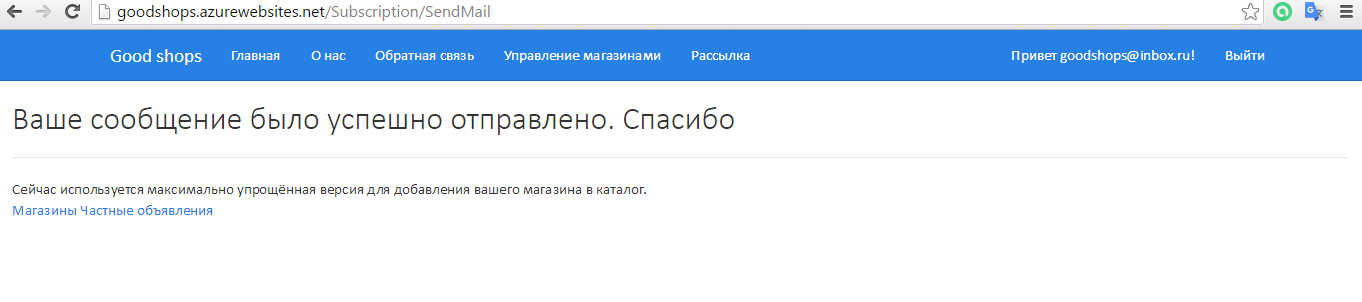


Рисунок 31 – Результат рассылки сообщений

Все подписчики получат отправленное администратором письмо. Результат представлен на рисунке 32.

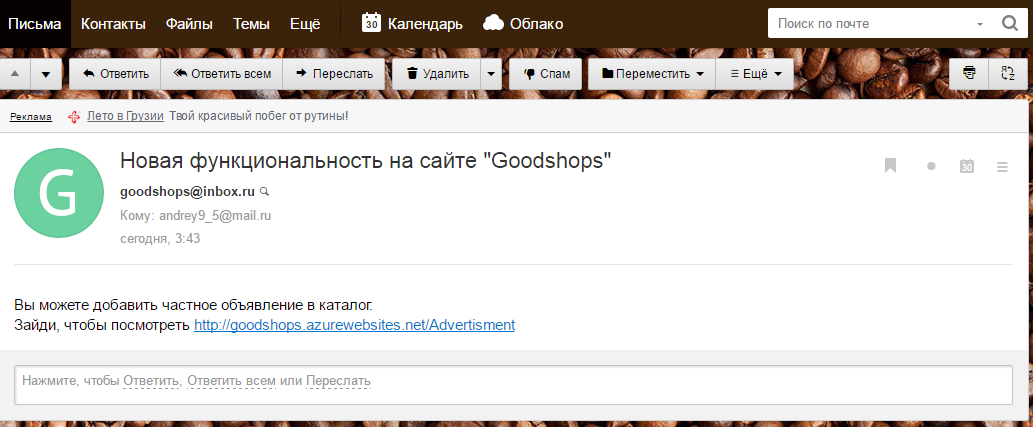


Рисунок 32 – Полученное письмо, отправленное администратором

Для того, чтобы оставить отзыв, либо задать интересующий вопрос, необходимо открыть страницу «Обратная связь» из главного меню сайта. Если пользователь авторизован, то поле «Адрес электронной почты» будет автоматически заполнено. Страница «Обратная связь» представлена на рисунке 33.

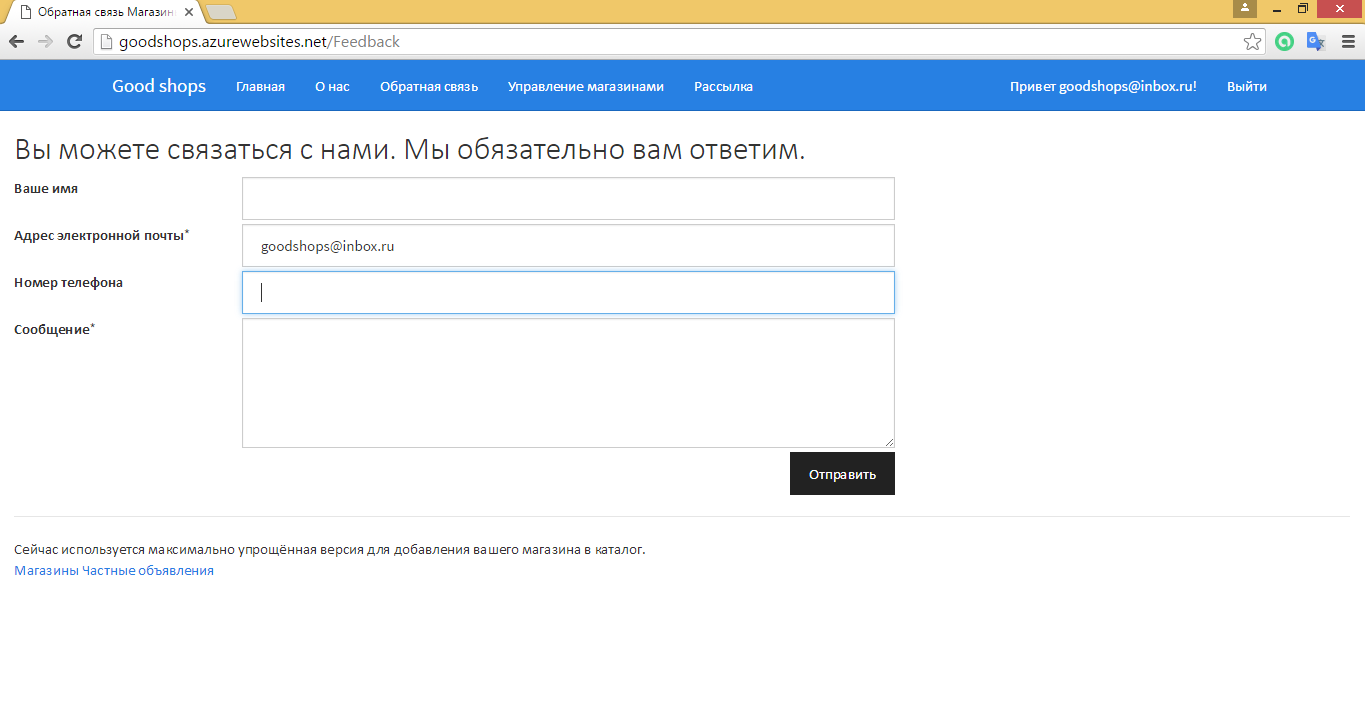


Рисунок 33 – Страница «Обратная связь»

Поля «Сообщение» и «Адрес электронной почты» являются обязательными для заполнения, если поля не будут заполнены, пользователь увидит ошибки, представленные на рисунке 34.

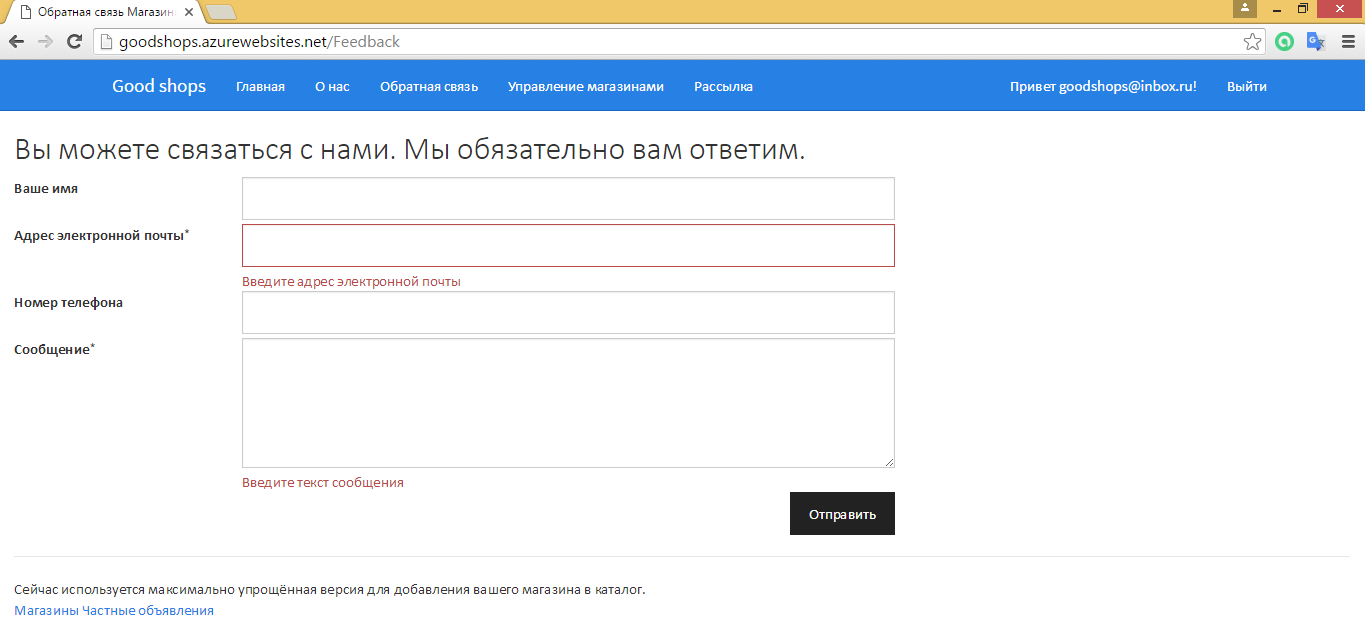


Рисунок 34 – Валидация на странице «Обратная связь»

После заполнения всех данных и нажатия на кнопку отправить, администратор перенаправится на страницу, представленную на рисунке 31.

На почту администратора сайта придёт отправленное пользователем письмо.

Для регистрации на сайте необходимо в главном меню нажать на элемент «Регистрация». Страница для регистрации пользователей представлена на рисунке 35.

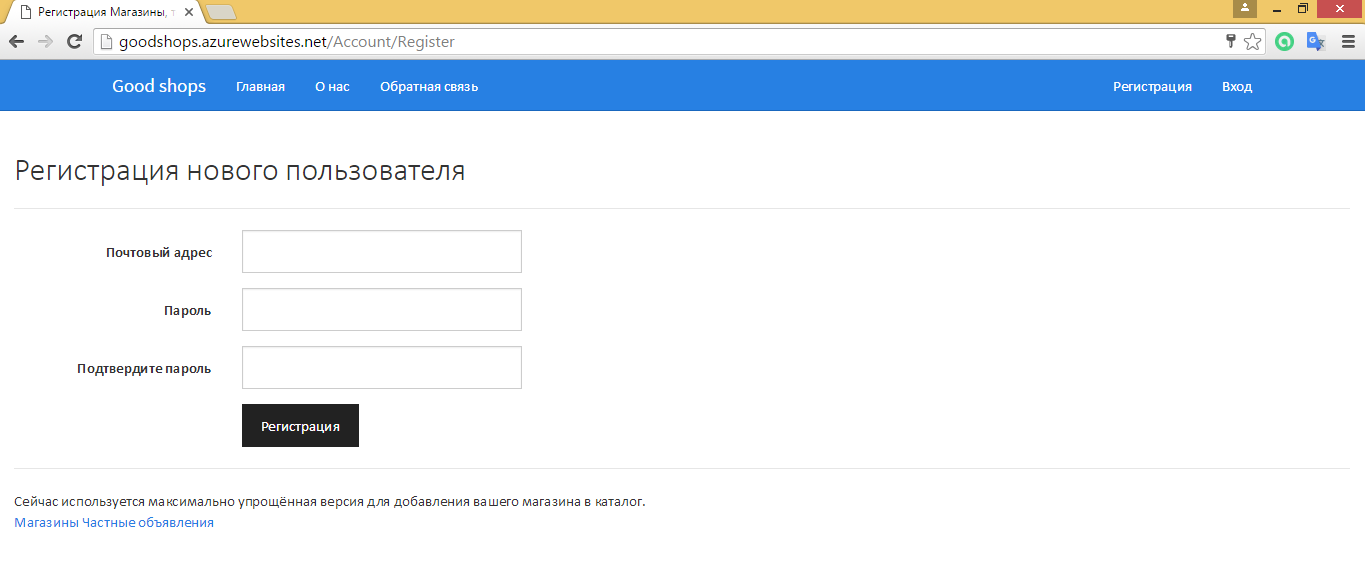


Рисунок 35 – Страница для регистрации пользователей

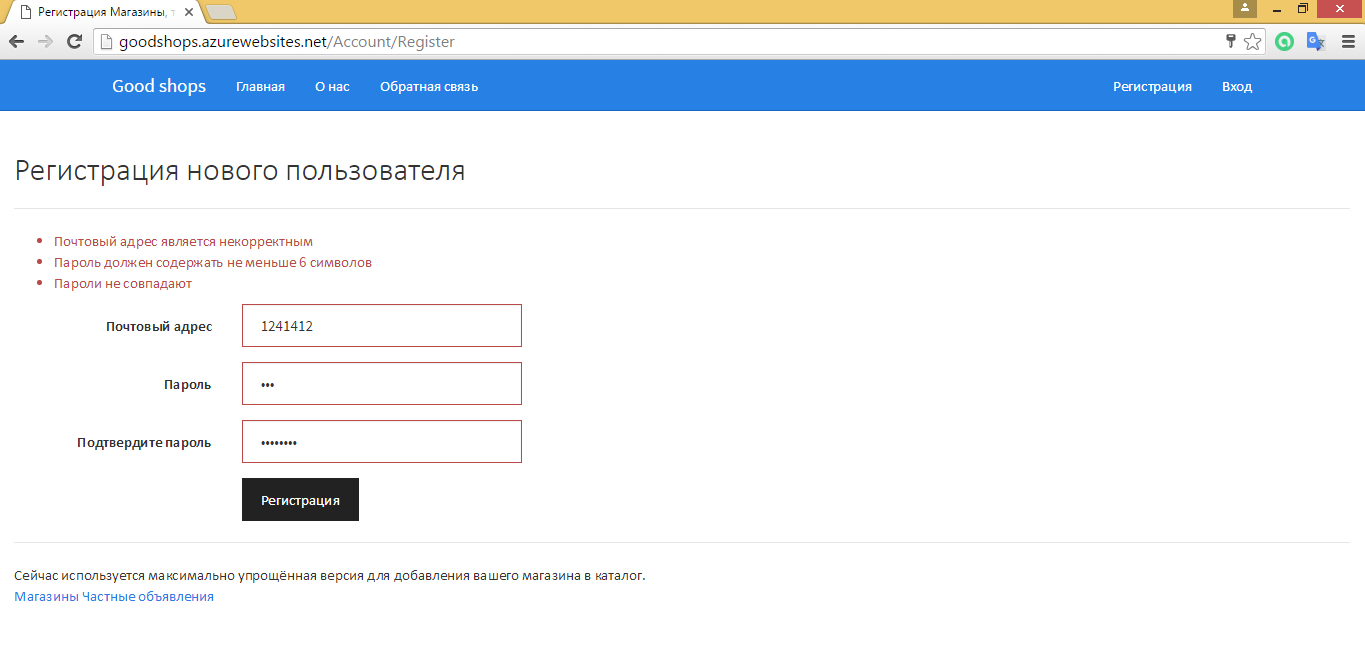
 Если форма будет заполнена некорректно, пользователь увидит соответствующие ошибки, примеры ошибок представлены на рисунке 36.

Рисунок 36 – Валидация на странице для регистрации пользователей

После того как пользователь корректно заполнит данные и нажмёт кнопку «Регистрация», для него будет создан аккаунт в системе. Имея введённые ранее электронную почту и пароль, пользователь сможет, имея эти данные залогиниться в системе.

Страница для входа на сайт представлен на рисунке 37.

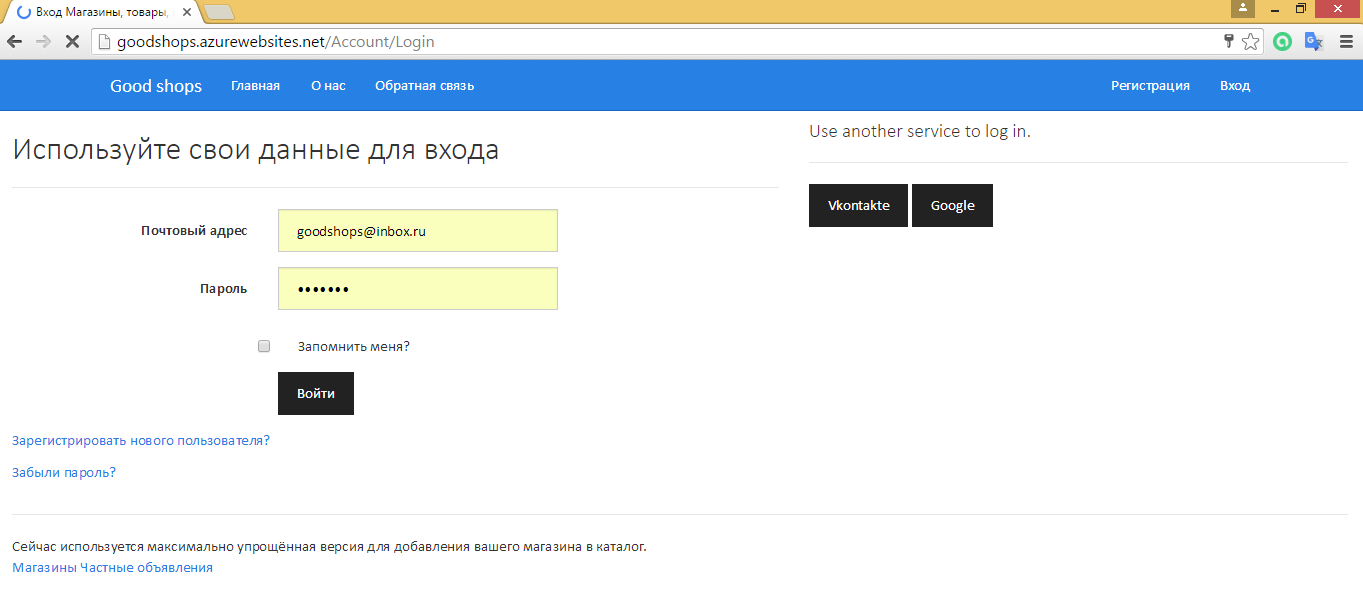


Рисунок 37 – Страница для входа пользователей на сайт

На сайте не обязательно проходить процедуру регистрации, если пользователь имеет аккаунт в vkontakte, он может нажать на странице «Вход» кнопку «vkontakte». После подтверждения аккаунта «vkontakte» пользователю будет предложено оставить такой же email, как и в социальной сети либо поменять его. Эта страница представлена на рисунке 38.

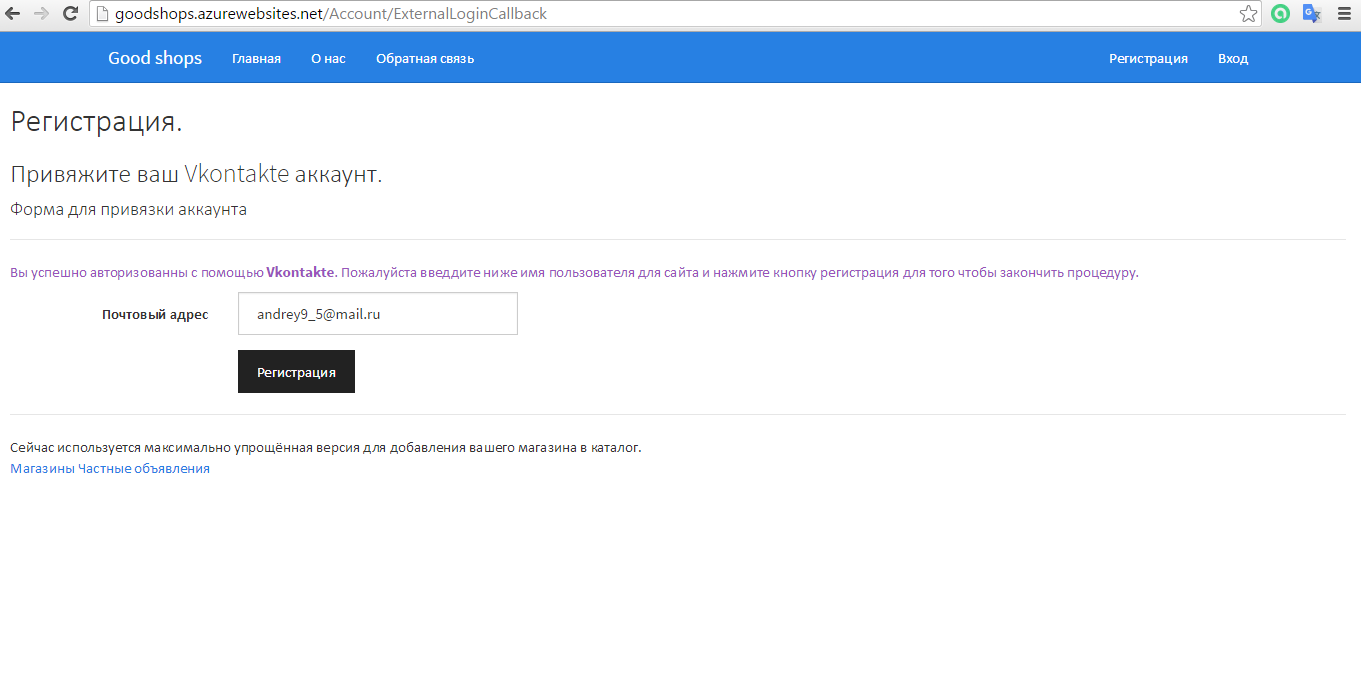


Рисунок 38 – Авторизация через «VKontakte»

Администратор может управлять заявками на создание магазина, а также блокировать или снимать блокировку с магазина. Данные функции представлены администратору на странице «Управление магазинами», представленной на рисунке 39.

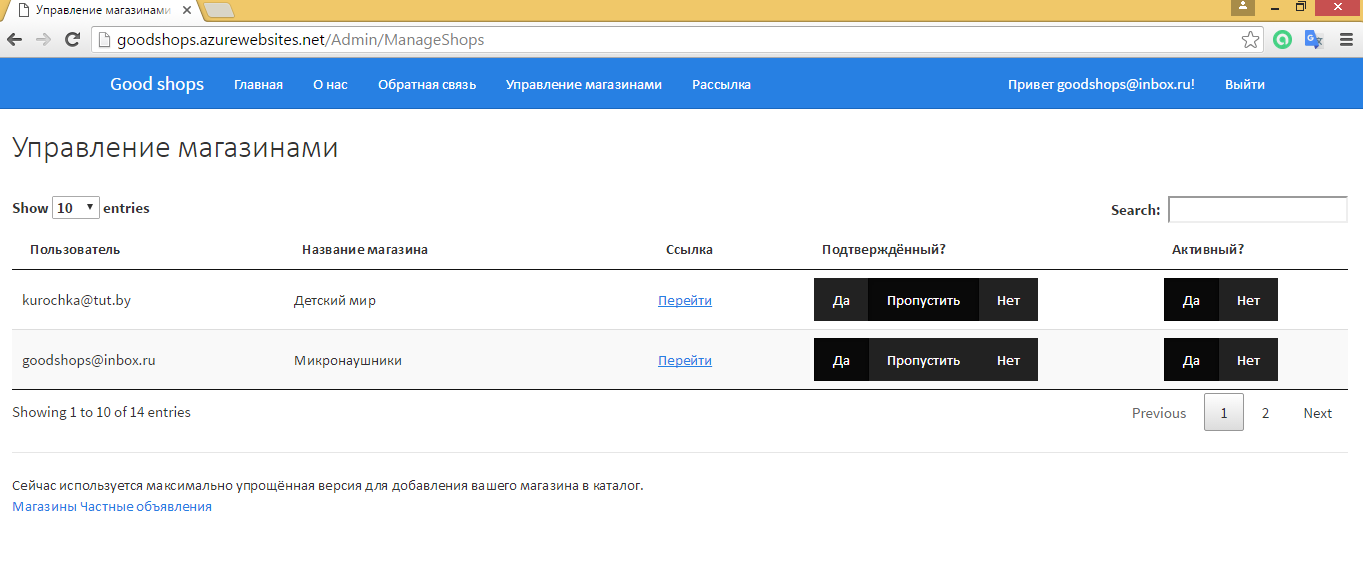


Рисунок 39 – Страница «Управление магазинами»

После того, как администратор выберет действие, которое необходимо выполнить, появится всплывающее окно, в котором необходимо подтвердить операцию. Окна для подтверждения действий представлены на рисунках 40, 41.

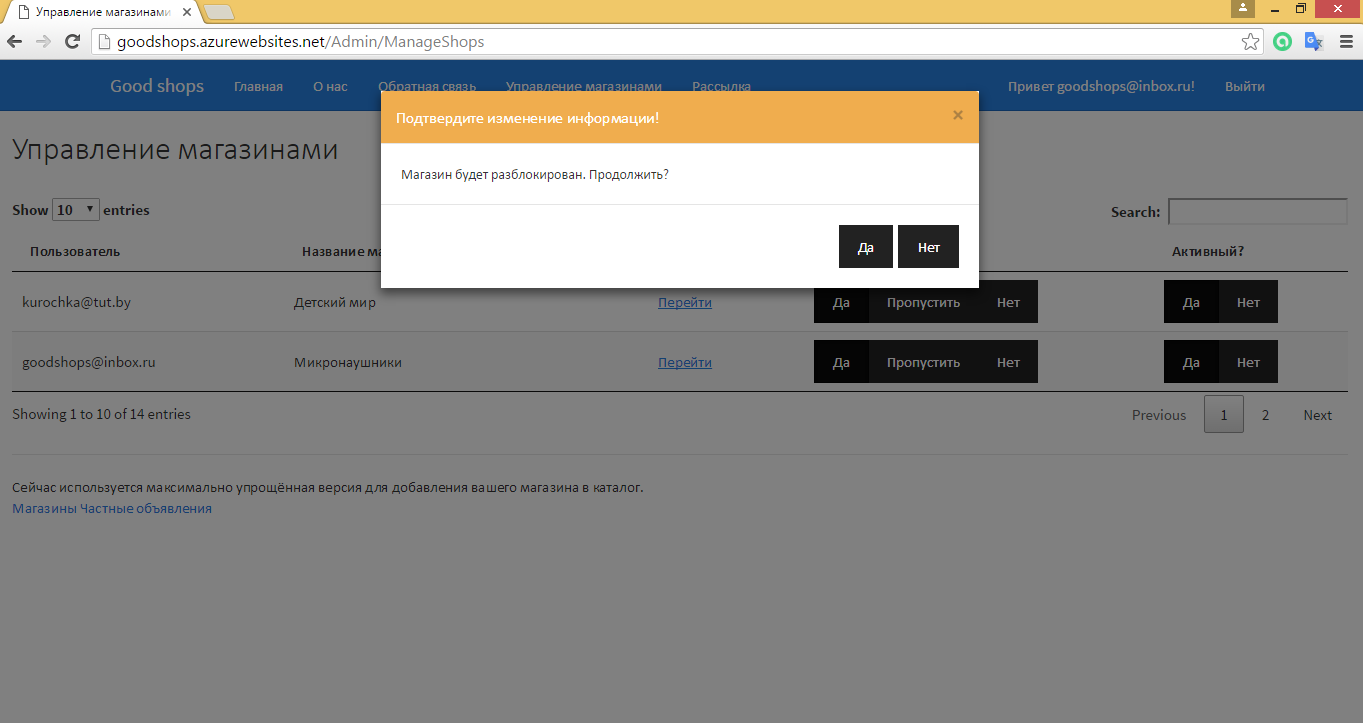


Рисунок 40 – Всплывающее окно для подтверждения действия «разблокировать магазин».

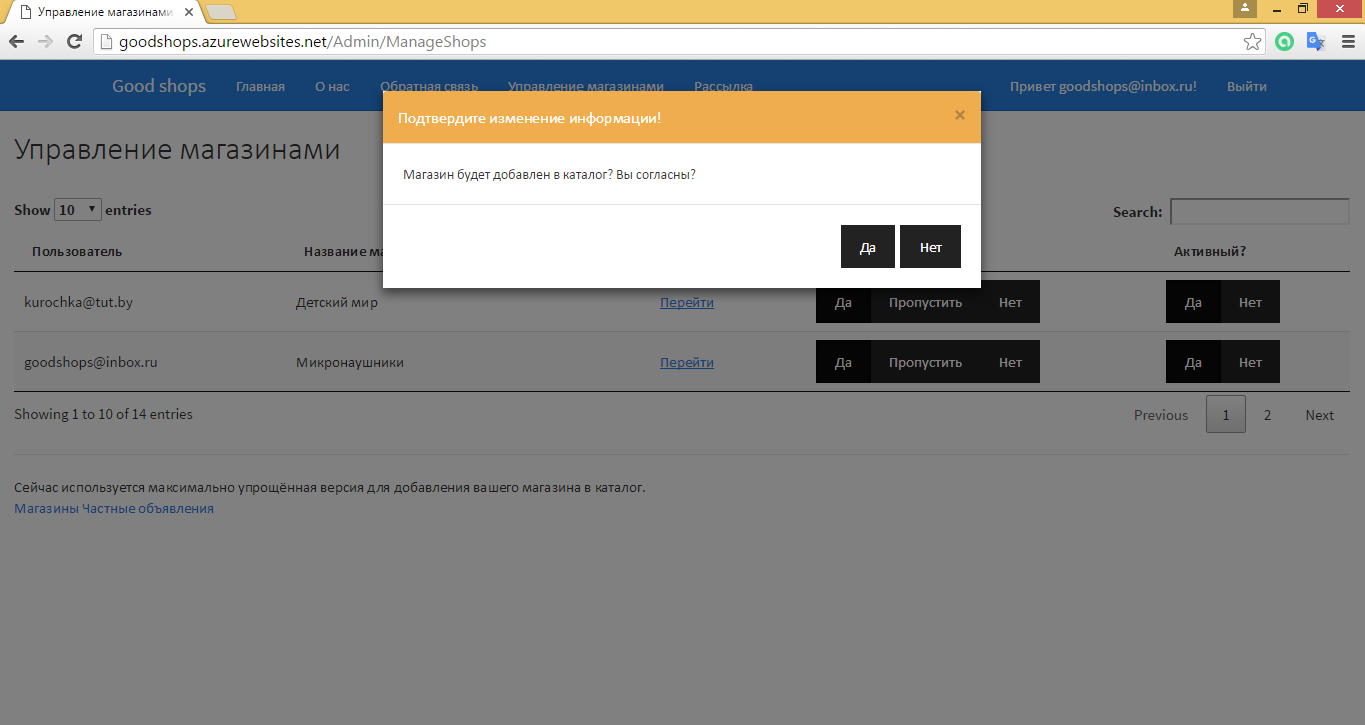


Рисунок 41 – Всплывающее окно для подтверждения действия «Добавить магазин в каталог»

После подтверждения операции с магазином администратор получит уведомление, представленное на рисунке 42.

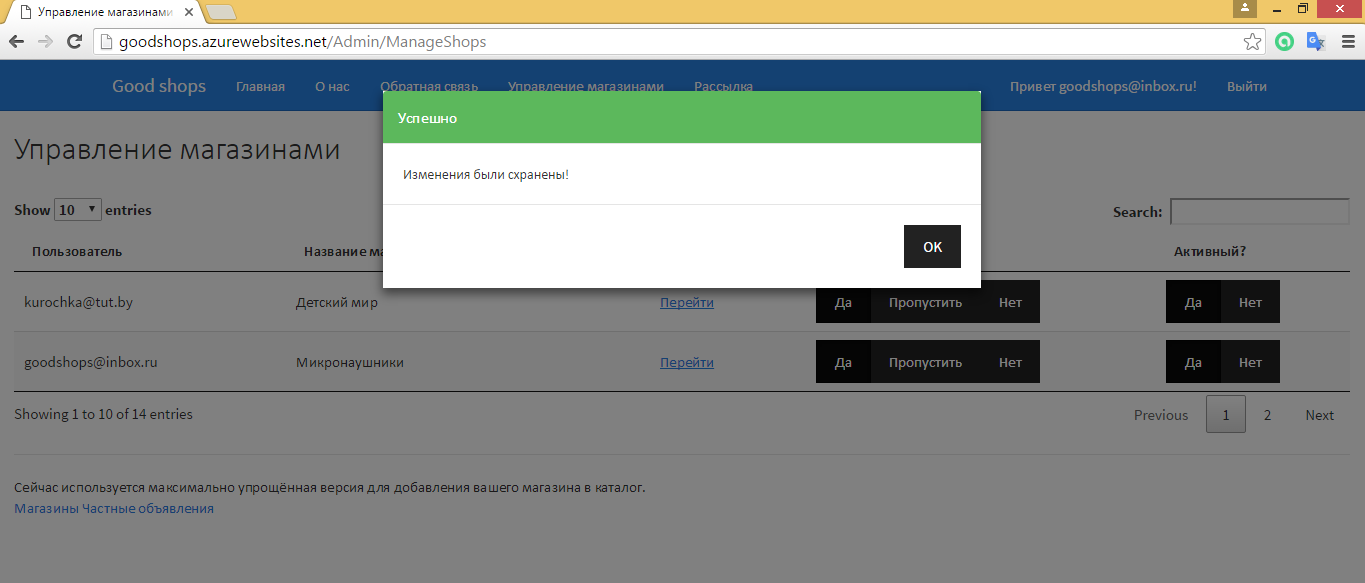


Рисунок 42 – Всплывающее окно об успешном выполнении операции

Если магазин является активным и разблокированным, то любой пользователь сможет его видеть на странице «Магазины».

4 Результаты

Выполненный объем работы по дипломному проекту обеспечил достижение поставленной перед разработчиком цели. Удалось добиться стабильной работы приложения при разных типах входных данных. Получены необходимые знания для расширения реализованной системы. Разработана техническая документация в виде данной пояснительной записки. Проведено тестирование программных модулей.

Заключение

Разработан запланированный модуль информационной системы. Продумана архитектура приложения и реализован весь намеченный в ходе планирования функционал. Реализовано централизованное хранилище и логика по хранению динамичных и статичных данных ИС.

Разработанная ИС «Goodshops App» обеспечивает следующие функциональные возможности:

* авторизация пользователей;
* обратная связь;
* управления заявками магазинов;
* подписка на обновления сайта;
* рассылка сообщений

В ходе выполнения дипломной 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.