**Курсовая работа «Кондитерский цех»**

Содержание

[Введение 4](#_Toc121480392)

[1 Теоретическая часть 9](#_Toc121480393)

[1.1 Выбор языка и среды программирования 9](#_Toc121480394)

[1.2 Разработка системы требований 13](#_Toc121480395)

[1.3 Разработка функциональной спецификации 15](#_Toc121480396)

[2 Практическая часть 17](#_Toc121480397)

[2.1 Разработка CRC-карточек 17](#_Toc121480398)

[2.2 Структурная и функциональная схемы 18](#_Toc121480399)

[2.3 Разработка диаграммы деятельности 20](#_Toc121480400)

[2.4 Разработка ER-диаграммы 21](#_Toc121480401)

[2.5 Разработка словаря данных 23](#_Toc121480402)

[2.6 Тестирование программного обеспечения 24](#_Toc121480403)

[Заключение 25](#_Toc121480404)

[Список литературы 26](#_Toc121480405)

Введение

Интернет в наши дни — это целая индустрия, быстро проникающая во все области человеческой деятельности. Огромное количество компаний во всем мире видят в Интернет большой коммерческий потенциал и возможность перевода своего бизнеса на качественно новый уровень. Благодаря постоянному совершенствованию технологий, скорость доступа растет, а цена становится всё более приемлемой. Интернет стал полноценной социальной средой, в которой сотни людей встречаются, общаются, покупают и продаются товары, делятся новостями, мнениями, получают необходимую им информацию.

Интернет развивается стремительно и сегодня большинство компаний уже имеют свои сайты. В настоящее время невозможно представить современный бизнес без использования Интернет-технологий. Впрочем, не только компании, но и частные лица, оказывающие те или иные услуги.

Иметь интернет-ресурс – это, скорее, необходимость. Статистика утверждает, что в среднем каждый человек ежедневно проводит в интернете не менее 3-х часов. Это если не говорить о тех, кто связан с ним по работе. Учитывая такие данные, компании стали реагировать соответствующим образом и вести бизнес через интернет. Люди оценили удобство сети. Ведь она позволяет вести бизнес, узнавать информацию, бронировать билеты и гостиницы, просто развлекаться. Возможности, предоставляемые такими ресурсами, невероятные.

Основные причины необходимости сайта для компании:

1. Лицо компании

Наличие собственного сайта существенно повышает имидж компании в глазах потенциальных клиентов и партнеров. Если его нет, к компании будут относиться не только настороженно, но и несерьезно. У клиентов сложится четкое впечатление, что компания не беспокоится о своем имидже и не следует веяниям времени.

2. Быстрый доступ к информации

Еще несколько лет назад, чтобы узнать подробности работы компании, уточнить, какие именно услуги она предоставляет, нужно было звонить или даже приходить в офис. Сделать это можно было лишь в рабочее время.

Сайт – это виртуальный офис, работающий круглосуточно и без выходных. В любой момент пользователь может зайти «в него» и получить нужную ему информацию:

* подробное описание товаров и услуг;
* контактные данные;
* время работы и т.д.

Это очень удобно для пользователей и полезно для самой компании.

3. Возможность заявить о себе

С помощью сайта вы сможете рассказать о себе, предоставляя пользователям всю возможную информацию, рассказывая, в чем ваши конкретные преимущества перед прочими компаниями.

Вероятно, потенциальные клиенты будут искать не конкретную компанию в интернете, а любую, предоставляющую аналогичные услуги, проанализировав их сайты и их предложения, выберут наиболее понравившийся вариант и обратятся к ним. Если же у компании нет этого сайта, то и о существовании компании они вероятно даже не узнают.

4. Инструмент ведения бизнеса

Сайт являет собой реальный и эффективный, действенный инструмент ведения бизнеса. На нем будет представлена вся важнейшая информация, которая только может понадобиться:

* клиентам;
* партнерам;
* поставщикам.

Ее наличие, четкое и понятное описание привлечет к компании больше заказчиков. Если же на сайте продумать все детали, предусмотреть форму обратной связи или сразу установить кнопку для заказа товаров/услуг, эффективность ресурса вырастет в разы.

5. Средство информации

Речь идет не только о новостных сайтах или порталах. Даже если это корпоративный ресурс, он позволит предоставлять пользователям актуальную и полезную информацию:

* о последних новостях;
* о проводимых акциях;
* о существующих скидках и т.д.

6. Недорогой, но эффективный способ рекламы

Как минимум 30% клиентов узнают о компании именно через сайт. Понятно, что ежегодно эта цифра будет только увеличиваться. Уже сегодня интернет имеет массу преимуществ перед более привычными СМИ, посредством которых ранее запускалась реклама.

Учитывая популярность интернета, неудивительно, что сайт окупит себя уже через несколько месяцев, а то и недель. Правильно созданный, интересный, наполненный важной информацией сайт быстро приносит своим владельцам не просто доход, а реальную, осязаемую прибыль.

Например, стоимость месячной аренды щита под размещение рекламы приблизительно равняется стоимости создания сайта. Но через месяц рекламу снимут, а сайт – останется. К тому же, количество откликов на информацию на сайте намного выше, чем на рекламу на щите – потому что на сайт приходит целевая аудитория.

Наличие сайта для кондитерского цеха поможет показать себя не любительской компанией, а профессиональной, которая серьезно настроена на развитие и рост своего дела.

Причины необходимости сайта для кондитерского цеха:

* с большей вероятностью компания будет воспринята как серьезный и надежный партнер;
* повышение доверия со стороны клиентов;
* увеличение дохода;
* проще рекламировать компанию;
* пассивный поиск клиентов;
* проще делать заказы со стороны клиентов;
* удобное хранение заказов и их истории;
* экономия времени и персонала;
* возможна дополнительная прибыль за счет размещения рекламы на сайте;
* возможность предоставления подробной информации о предлагаемых товарах;
* возможность очень быстро оказывать информационную поддержку своим клиентам и отвечать на интересующие их вопросы (иногда разговаривая по телефону, сотрудник фирмы может не найти с ходу нужного ответа, а клиент упустит из виду какой-нибудь важный момент; сообщение в виде текста способствует понятной и подробной формулировке как вопроса, так и ответа);
* возможность привлечения большего объёма целевой аудитории (как правило, в сети Интернет люди ищут нужную информацию, используя поисковые системы, где они конкретно указывают то, что их интересует;

Интернет находит своё применение во многих сферах, не обходит стороной и деятельность коммерческих фирм, в том числе и кондитерского цеха. Создание сайта является для кондитера большим и серьезным шагом на пути от любителя к профессионалу. Персонально спроектированный и реализованный сайт для компании в разы упростит маркетинговую работу фирмы и увеличит кол-во заказов, в чем и заключается актуальность проекта.

Цель курсовой работы: разработать сайт для кондитерского цеха.

Для достижения цели необходимо решить ряд задач:

* исследовать изучаемую область и ее особенности;
* изучить технологии создания веб сайта;
* составить систему требований;
* провести анализ аналогичных сайтов других кондитерских;
* разработать макет и структуру сайта;
* произвести UML – проектирование сайта;
* спроектировать базу данных и диаграммы «Сущность - связь»;
* создать графический макет веб-сайта;
* разработать сайт;
* протестировать сайт;
* составить программную документацию.

Роль разработчика заключается в разработке документации, дизайна, схем, базы данных, самого сайта, его запуска, сопровождении и презентации.

Практическая значимость разрабатываемого web-сайта может заключаться в его использовании в качестве реального сайта начинающей кондитерского цеха после модифицирования под конкретные требования компании.

1 Теоретическая часть

1.1 Выбор языка и среды программирования

Для реализации проекта необходимо выбрать инструменты, с помощью которых будет производится разработка. Для начала необходимо определить, будет ли разрабатываться сайт вручную, с использованием CMS или фреймворков.

Использование CMS, таких как WordPress, Joomla и др. не целесообразно в рамках данного проекта. CMS занимают довольно много места, потребуется дополнительное время на изучения их архитектур, скорость работы снижена за счет обращений к БД, высокое потребление системных ресурсов, сложно адаптировать для нетипичных задач, в многих CMS периодически находят уязвимости и они постоянно требуются обновлений.

Разработка на фреймворках, таких как Laravel, Yii и др. отнимает много времени, так как большинство функционала придётся создавать самостоятельно с нуля; для того, чтобы разбираться с готовым кодом, написанным другим программистом, потребуется большое количество времени и практика, поэтому без достаточного опыта создать большой сайт не получится. Помимо этого, скорость загрузки некоторых приложений может быть снижена за счет объема фреймворка, а из-за необходимости предварительно скомпилировать код для многократного использования, требуется больше времени для тестирования.

Подводя итог вышенаписанному, наиболее удачным вариантом будет самостоятельная разработка проекта без готовых решений.

Разработка делится на 2 основных этапа: Frontend и Backend.

Frontend-разработка — это создание пользовательского интерфейса на клиентской стороне веб‑сайта или приложения. Это всё, что видит пользователь, когда открывает веб-страницу, и с чем он взаимодействует: кнопки, баннеры и анимация. Frontend сайта или веб-приложения можно разделить на три большие части:

* HTML (HyperText Markup Language) — язык разметки документов, на котором создают структуру страницы: заголовки, абзацы, списки и так далее.
* CSS (Cascading Style Sheets) — язык для описания и стилизации внешнего вида документа. Благодаря CSS-коду браузер понимает, как именно отображать элементы. Например, CSS задаёт цвета и параметры шрифтов, определяет, как будут располагаться разные блоки сайта. Ещё он позволяет воспроизводить документ в разных видах: для печати (обычной или шрифтом Брайля), вывода на экран или для чтения скринридером.
* JavaScript — это язык, который оживляет веб-страницы. Его задача — реагировать на действия пользователя, обрабатывать клики мышки, перемещения курсора, нажатия клавиш. Ещё он посылает запросы на сервер и загружает данные без перезагрузки страницы, позволяет вводить сообщения и многое другое.

Для организации структуры и удобства разработки используется такой инструмент, а если быть точнее, то «сборщик» под названием Gulp.

Gulp — менеджер для организации и выполнения задач при разработке приложений с использованием платформы Node.js Он может выполнять разные задачи: формировать файлы стилей, объединять и минифицировать файлы, оптимизировать изображения, транспилировать код и т. д. При выполнении этих задач происходит взаимодействие с рабочим окружением: обратиться к файловой системе, создать или сохранить итоговый файл, запустить процесс локального web-сервера, показать результат в браузере.

Таким образом разработка Frontend части будет производится с помощью HTML, CSS и JavaScript. В качестве дополнительного инструмента разработки будет использован сборщик GULP.

Backend — это разработка бизнес-логики продукта (сайта или веб-приложения). Backend отвечает за взаимодействие пользователя с внутренними данными, которые потом отображает frontend. Другими словами, это то, что скрыто от глаз пользователя и происходит вне его браузера и компьютера.

В качестве языков программирования, на которых может производится разработка, представлены PHP, Python и Java.

Язык Java является надежным, но разработка на нем потребует большего количества времени. Он хорошо показывает себя в больших проектах, но для реализации данного проекта целесообразнее использовать более простой язык. Python за счет своей универсальности довольно обширен, поэтому только его изучение потребует много времени. Т.о. в качестве языка программирования выбран язык PHP.

Главными преимуществами языка PHP являются простота в изучении и довольно высокая производительность, но, помимо этого, можно отметить следующие плюсы данного языка:

* мощность и гибкость, отлично подходит как для небольших, так и для крупных проектов;
* простота изучения;
* большое сообщество, следовательно, можно найти много готовых решений и инструкций;
* имеет большое количество дополнительных расширений и библиотек, которые увеличивают его функциональность;
* является полностью бесплатным и распространяется со свободной лицензией, поэтому его смело могут применять как частные лица, так и коммерческие организации;
* не требователен и может применять на всех известных видах серверов.

Backend часть не может быть реализована без работы с Базой Данных. Лучшим выбором является СУБД MySQL за счет ее бесплатности, высокой скорости обработки данных, популярности, универсальности, поддержки SQL (что обеспечивает высокий уровень кроссплатформенности данных и кода), использования системы привилегий, хеширования паролей и работы в связке с (выбранным для этого проекта) языком PHP.

Для более комфортной и быстрой работы следует использовать программу PhpMyAdmin.

PhpMyAdmin - веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL. PhpMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных.

Некоторые из ключевых преимуществ, которые предоставляет phpMyAdmin:

* Веб−интерфейс - Будучи веб-интерфейсом, пользовательский интерфейс phpMyAdmin доступен с помощью веб-браузера, и этот интерфейс доступен на всех платформах, где может работать веб-браузер.
* Графический интерфейс phpMyAdmin предоставляет графический интерфейс для запуска команд SQL и выполнения операций SQL и делает его довольно простым в использовании по сравнению с консольными редакторами sql.
* Интерфейс скрипта phpMyAdmin предоставляет интерфейс скрипта для запуска PHP-скриптов для подключения к базам данных и выполнения пользовательских операций.
* Многосерверный phpMyAdmin позволяет управлять несколькими серверами одновременно.
* Форматы резервного копирования phpMyAdmin позволяет создавать резервные копии баз данных в различных форматах, таких как XML, CSV, SQL, PDF, OpenDocument Text, Excel, Word, электронная таблица и т.д.
* Простой в использовании интерфейс phpMyAdmin позволяет легко создавать и запускать сложные запросы, создавать и редактировать функции, триггеры и т. Д.

В качестве среды разработки выбран текстовый редактор Atom, за счет быстрого написания кода с умным авто дополнением, удобным разделением интерфейса редактирования для одновременной работы с несколькими файлами или проектами, возможности установки расширений с помощью встроенного менеджера пакетов, а также за встроенную интеграцию с Git и GitHub.

Таким образом разработка frontend части будет производится на языках HTML, CSS, JS при помощи GULP. А backend – на языке программирования PHP, СУБД – MySQL с использованием программы PhpMyAdmin. Среда разработки –редактор кода Atom.

1.2 Разработка системы требований

Для составления системы требованием необходимо четко определить проблему для дальнейшего ее решения. Структурирование проблемы приведено в таблице 1

1. Структурирование проблемы

|  |  |
| --- | --- |
| Элемент | Описание |
| Проблема | Как увеличить доходы компании и автоматизировать покупку товара |
| Воздействует на что (кого) и  результатом чего является | Проблема оказывает влияние на Заказчиков и Компанию.  От качества реализации зависит репутация компании, впечатление Заказчиков от качества выполнения работы и, следовательно, доходы кондитерского цеха. |
| Выигрыш от решения может  состоять в следующем | Предлагаемое решение: Создание веб-сайта, где Заказчик сможет просмотреть и заказать товар. Преимущества от данного решения:  - повышения доверия со стороны заказчика;  - повышение репутации компании за счет разработки сайта;  - удобства рекламирования кондитерского цеха;  - удобство совершения заказов без прямого посещения цеха или звонков в компанию. |

Таким образом необходимо разработать современный, красивый и практичный сайт для того, чтобы он мог заинтересовать потенциальных клиентов.

Определение корневых причин, лежащих в основе проблемы:

* У компании обязан быть собственный сайт, который будет выступать ее лицом;
* Рекламная компания проводится наиболее эффективно и выгодно за счет маркетинга сайта;
* Клиенты, совершающие заказ, должны предварительно увидеть товары;
* Обращение в поддержку занимает слишком много времени;
* Частые звонки по телефону для уточнения информации подходят не всем клиентам, сотрудники могут заниматься более продуктивной работой, нежели периодически отвечать на звонки.

Экономический покупатель системы – Руководитель Кондитерского цеха.

Пользователями системы являются Клиенты и Кондитеры компании. Помимо них, система может оказать косвенное влияние на потенциальных клиентов компании, повышая репутацию кондитерского цеха.

После того, как система будет разработана, представлена и развернута, проводится тестирование от сотрудников компании, после которого система оценивается. Других внешних или внутренних пользователей системы, чьи потребности следует учесть нет.

После проведения тестирования и сдачи системы, ее сопровождением займутся сотрудники компании-заказчика. Управление системой передается администратору. Система имеет связь с Базой Данных, откуда и получает информацию.

Ограничения для системы:

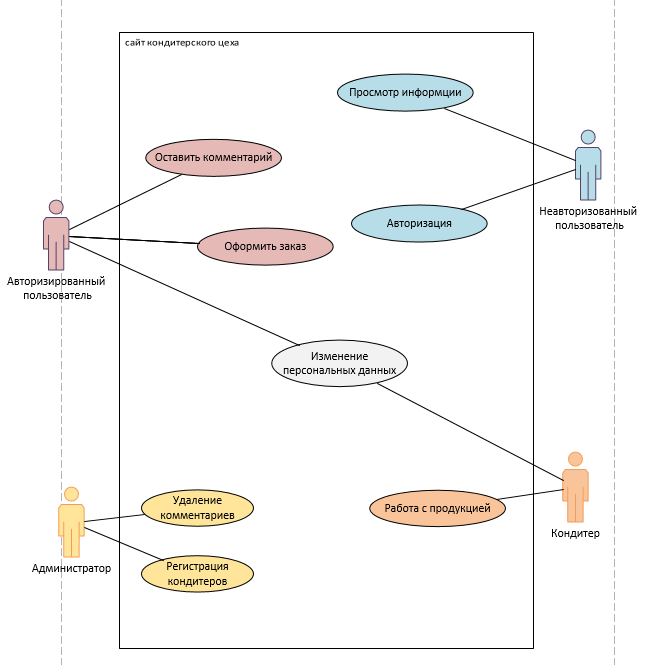
* обеспечение непрерывной работы сайта;
* соответствуя требованиям безопасности, необходимо шифровать пароли в Базе Данных;
* разработка продукта в рамках законодательства о «Защите персональных данных» (Федеральный закон "О персональных данных" от 27.07.2006 N 152-ФЗ)

1.3 Разработка функциональной спецификации

Диаграмма прецедентов — это наиболее общее представление функционального назначения системы. На ней применяются два типа основных сущностей: варианты использования и действующие лица, между которыми устанавливаются следующие основные типы отношений:

* ассоциация между действующим лицом и вариантом использования;
* обобщение между действующими лицами;
* обобщение между вариантами использования;
* зависимости (различных типов) между вариантами использования.

Диаграмма прецедентов представлена на рисунке 1

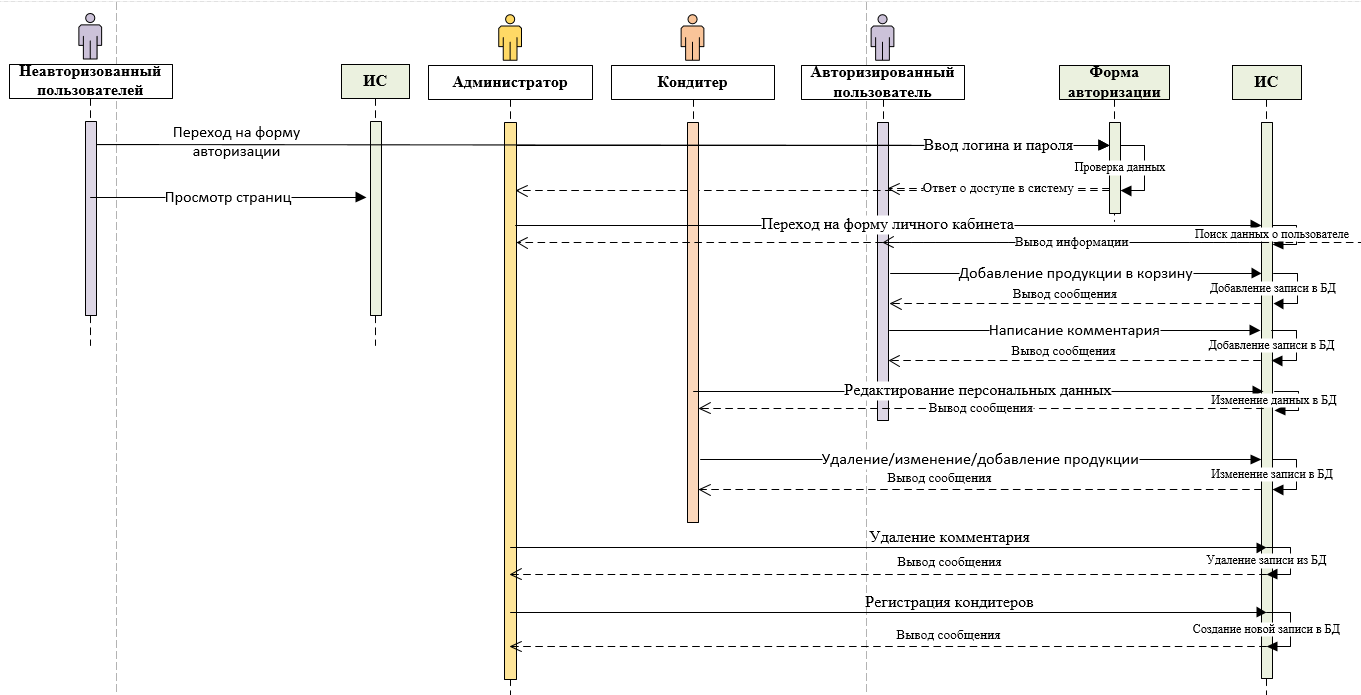


1. Дигамма прецедентов

Первый актер – Неавторизованный пользователь. Это любой посетитель сайта. Он может просматривать информацию на сайте и авторизоваться. После авторизации становится авторизированным пользователем. Авторизированный пользователь может редактировать свои персональные данные, оставлять комментарии на продукции и заказывать продукцию.

Диаграмма последовательности — диаграмма, на которой показаны взаимодействия объектов, упорядоченные по времени их проявления. Используется в языке UML. По сути, это запись протокола конкретного сеанса работы системы (или фрагмента такого протокола). В объектно-ориентированном программировании самым существенным во время выполнения является посылка сообщений взаимодействующими объектами. Именно последовательность посылки сообщений отображается на данной диаграмме.

Диаграмма последовательностей системы представлена на рисунке 2



1. Диаграмма последовательностей системы

На диаграмме последовательностей показаны возможности всех ролей. Присвоение другой роли Неавторизованному пользователю происходит после авторизации. Регистрация пользователей роли Кондитер производится Администратором.

2 Практическая часть

2.1 Разработка CRC-карточек

Одним из способов проектирования программного обеспечения является метод CRC-карточек. Целью использования данного метода является исследование взаимодействия объектов системы, и как результат - более ясное понимание структуры программы.

На этом этапе необходимо выявить классы, которые необходимо будет создать в программе для реализации системы. Для понимания работы проектируемой информационной системы достаточно выделить следующие классы:

1. клиент;
2. кондитер;
3. администратор;
4. веб-сайт;

CRC-карты относительно выделенных классов приведены в таблицах 2-6

1. CRC-карта для класса Пользователь

|  |  |
| --- | --- |
| Клиент | |
| 1. Посещает веб-сайт; 2. Переходит на страницы сайта; 3. Переходит на форму авторизации; 4. Вводит логин, пароль и нажимает на кнопку вход; 5. Переходит на страницу личного кабинета; 6. Просматривает собственную информацию, при желании может изменить её; 7. Переходит на страницу продукции; 8. Добавляет товар в корзину; 9. Оформляет заказ. | Веб-сайт |

1. CRC-карта для класса Сотрудник

|  |  |
| --- | --- |
| Кондитер | |
| 1. Посещает веб-сайт; 2. Переходит на форму авторизации; 3. Вводит логин, пароль и нажимает на кнопку вход; 4. Переходит на страницу личного кабинета; 5. Просматривает собственную информацию, при желании может изменить её; 6. Добавляет/изменяет/удаляет продукцию; | Веб-сайт |

1. CRC-карта для класса Администратор

|  |  |
| --- | --- |
| Администратор | |
| 1. Посещает веб-сайт; 2. Переходит на форму авторизации Дочернего сайта; 3. Вводит логин, пароль и нажимает на кнопку вход; 4. Переходит на страницу личного кабинета; 5. Регистрирует кондитеров; 6. Удаляет комментарии; | Веб-сайт,  Кондитер |

1. CRC-карта для класса Веб-сайт

|  |  |
| --- | --- |
| Веб-сайт | |
| 1. Отображает основную информацию; 2. Соединяется с БД; 3. Добавляет продукцию в корзину; 4. Регистрирует заказ. | Клиент,  Сотрудник,  Администратор |

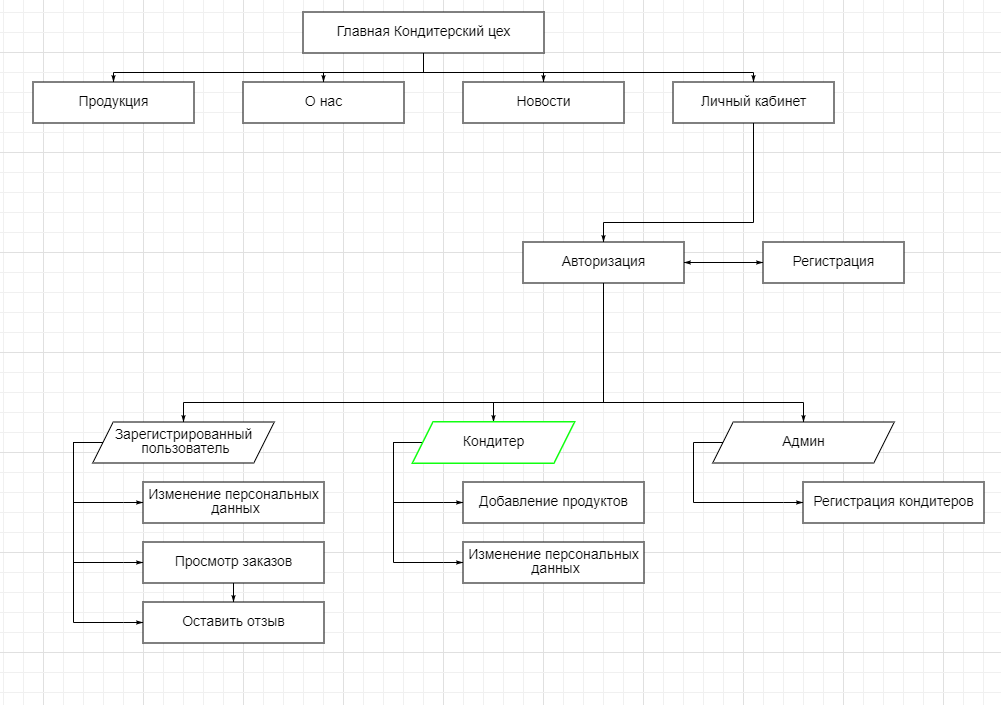
Таким образом выявлены 4 класса, которые необходимо создать в программе для реализации системы.

2.2 Структурная и функциональная схемы

Структурная схема отражает состав и взаимодействие по управлению частей разрабатываемого программного обеспечения, определяется архитектурой разрабатываемого ПО. Разработка структурной схемы программы выполняется методом пошаговой детализации.

Структурные схемы пакетов программ разрабатывают для каждой программы пакета по отдельности, поскольку организация программ в пакеты не предусматривает передачи управления между ними. Компонентами структурной схемы программной системы или программного комплекса могут служить программы, подсистемы, базы данных, библиотеки ресурсов и т. п.

Структурная схема сайта «Кондитерский цех» приведена на рисунке 3

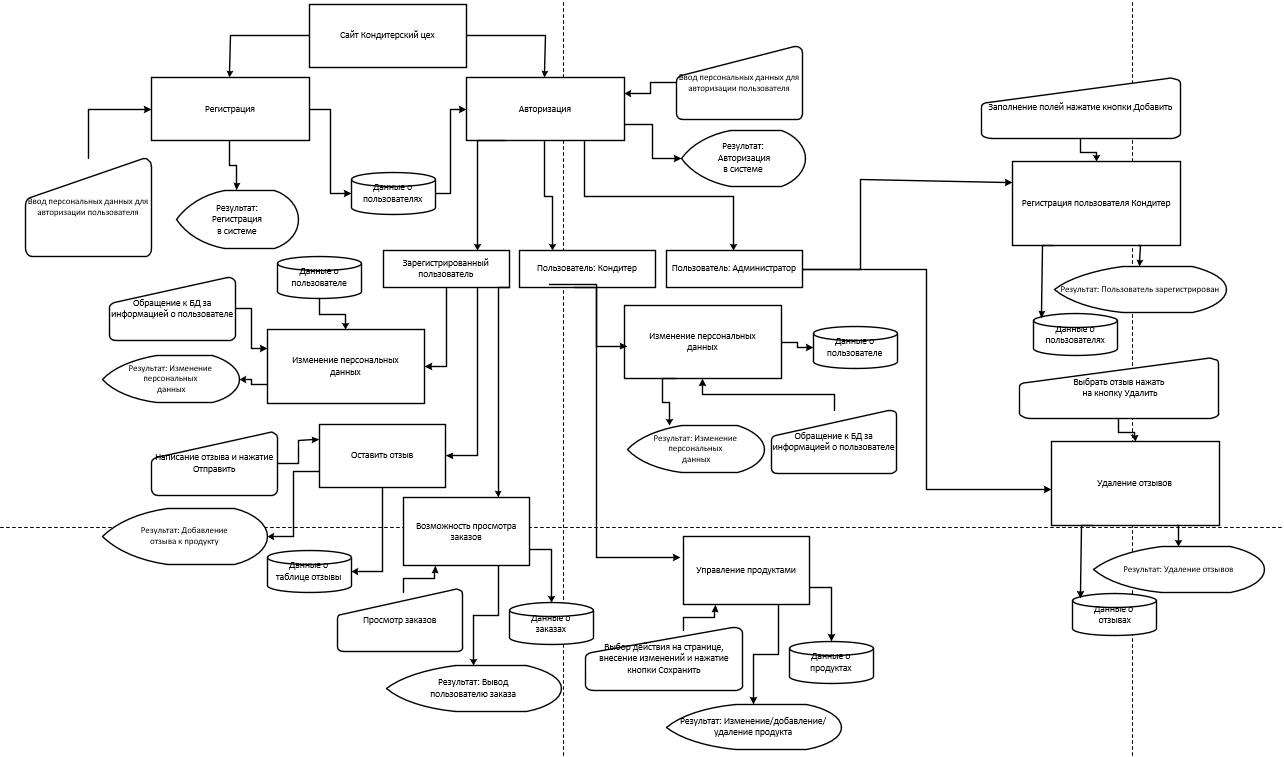


1. Структурная схема

На структурной схеме показаны, какие на сайте существуют страницы. Страница личного кабинета переносит пользователя на страницу авторизации при первом посещении. Если пользователь еще не зарегистрирован, он может это сделать. После авторизации у пользователей, в зависимости от их ролей, появляются дополнительные возможности, кроме просмотра контента.

Функциональная схема (ГОСТ 19.701—90) — это схема взаимодействия компонентов программного обеспечения с описанием информационных потоков, состава данных в потоках и указанием используемых файлов и устройств. Функциональные схемы более информативны, чем структурные.

Функциональная схема сайта Кондитерский цех приведена на рисунке 4



1. Функциональная схема

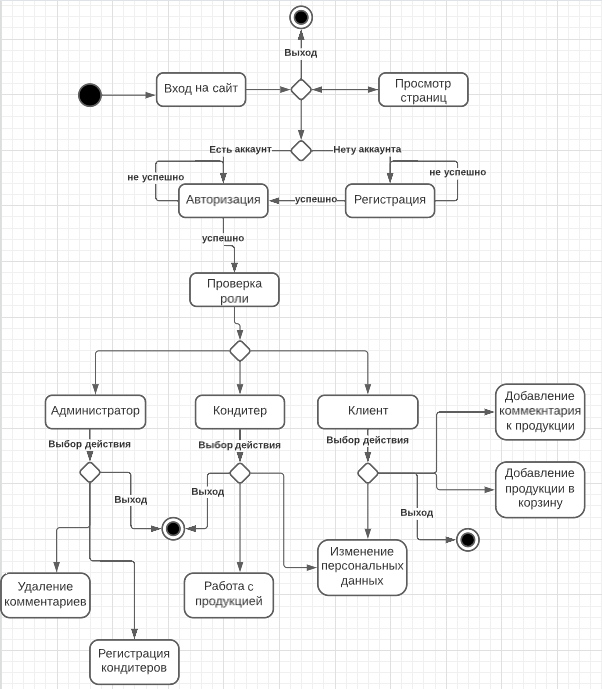
Из схемы видно, каким образом функции взаимодействуют с базой данных.

2.3 Разработка диаграммы деятельности

Для моделирования процесса выполнения операций в языке UML используются так называемые диаграммы деятельности. На этапе анализа требований и уточнения спецификаций диаграммы деятельностей позволяют конкретизировать основные функции разрабатываемого программного обеспечения.

Под деятельностью в данном случае понимают задачу (операцию), которую необходимо выполнить вручную или с помощью средств автоматизации. В теоретическом плане диаграммы деятельности являются обобщенным представлением алгоритма, реализующего анализируемый вариант использования.

Диаграмма деятельности представлена на рисунке 5



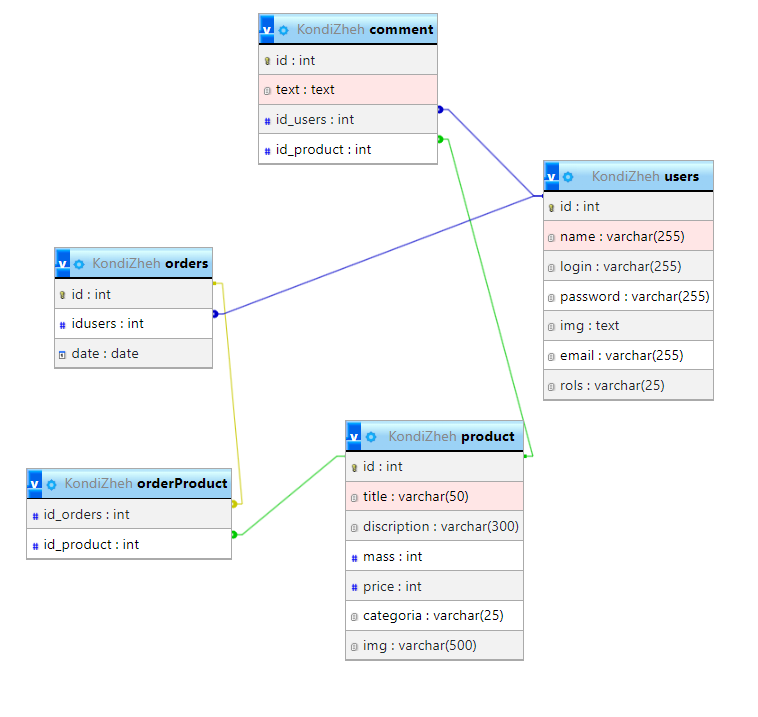
1. Диаграмма деятельности ИС «Кондитерский цех»

Главное предназначение диаграммы состояний — описать возможные последовательности состояний и переходов, которые в совокупности характеризуют поведение элемента модели в течение его жизненного цикла.

2.4 Разработка ER-диаграммы

ER-диаграмма — это разновидность блок-схемы, где показано, как разные «сущности» связаны между собой внутри системы. ER-диаграммы чаще всего применяются для проектирования и отладки реляционных баз данных в сфере образования, исследования и разработки программного обеспечения и информационных систем для бизнеса. ER-диаграммы полагаются на стандартный набор символов, включая прямоугольники, ромбы, овалы и соединительные линии, для отображения сущностей, их атрибутов и связей. Эти диаграммы устроены по тому же принципу, что и грамматические структуры: сущности выполняют роль существительных, а связи — глаголов.

На рисунке 6 представлена ER-диаграмма базы данных информационной системы «Кондитерский цех»



1. ER-диаграмма БД ИС «WEBEX»

На ER-диаграмме представлены основные таблицы ИС. Таблица users содержит данные пользователей, в таблице comment содержатся комментарии. Таблица product содержит данные о продуктах, orders о заказах, orderProduct – таблица связи двух таблиц.

2.5 Разработка словаря данных

Для каждого выходного документа был создан локальный СД. В результате проведения анализа перечня данных из входных документов и перечня данных из СД – определился общий СД, который содержит определение для всех данных из СД источников данных.

Словарь данных представлен на таблицах 7-12:

1. «Пользователи»

|  |  |  |  |
| --- | --- | --- | --- |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| **users** | | | |
| Первичный | id | Да | Номер пользователя |
|  | name | Да | Имя пользователя |
|  | login | Да | Логин пользователя |
|  | password | Да | Пароль |
|  | img | Нет | Аватар пользователя |
|  | email | Да | Почта |
|  | rols | Нет | Роль |

1. «Продукция»

|  |  |  |  |
| --- | --- | --- | --- |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| **product** | | | |
| Первичный | id | Да | Номер продукции |
|  | title | Да | Название |
|  | discription | Да | Описание |
|  | mass | Да | Масса |
|  | price | Да | Цена |
|  | categoria | Да | Категория |
|  | img | Нет | Картинка |

1. «Заказ»

|  |  |  |  |
| --- | --- | --- | --- |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| **orders** | | | |
| Первичный | id | Да | Номер заказа |
|  | date | Да | Дата заказа |
| Внешний | id\_users | Да | Внешний ключ к таблице «Пользователи» |

1. «Заказ-продукция»

|  |  |  |  |
| --- | --- | --- | --- |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| **orderProduct** | | | |
| Внешний | id\_orders | Да | Внешний ключ к таблице «Заказы» |
| Внешний | id\_product | Да | Внешний ключ к таблице «Продукты» |

1. «Комментарии»

|  |  |  |  |
| --- | --- | --- | --- |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| **comment** | | | |
| Первичный | id | Да | Номер комментария |
| Внешний | id\_users | Да | Внешний ключ к таблице «Пользователи» |
| Внешний | id\_product | Да | Внешний ключ к таблице «Продукция» |
|  | text | Да | Комментарий |

Веб-сайт взаимодействует с БД, большинство функционала становится доступным после авторизации.

2.6 Тестирование программного обеспечения

Заключение

Список литературы

* https://www.bibliofond.ru/view.aspx?id=877816
* https://vc.ru/flood/41366-kakie-specialisty-uchastvuyut-v-sozdanii-i-prodvizhenii-sayta
* https://vuzlit.com/385056/znachimost\_sayta