УДК 681.3.06

Палий Диана Николаевна

«Разработка веб-приложения для кулинаров “CookTasty”»

Квалификационная работа на степень «Бакалавр»  
Таганрог: ТТИ ЮФУ, 2013 г.- 80 с.

АННОТАЦИЯ

В дипломном проекте разработано веб-приложение для кулинаров «CookTasty».

«CookTasty» - это кулинарный сайт самых разнообразных рецептов для обычных пользователей сети Интернет. Цель проекта – помочь как тем, кто не разбирается в готовке, но хочет питаться правильно и вкусно, как и тем, кому просто требуется помощь, совет или мнение других людей.

«CookTasty» порадует своей простотой, яркостью, разнообразием и полнотой информации.

Так же в проекте проанализированы аспекты экономической эффективности предложенного проекта, безопасности и экологичности разработки, а так же её социальной значимости.

UDK 681.3.06

Paliy D.N.

«Development of web-based applications for cooks “CookTasty” »

Degree project –

TTI SFU, 2013 y. - 80 p.

**ABSTRACT**

There is a web-based application for culinary «CookTasty» in degree project.

«CookTasty» - it's a wide variety of culinary recipes for ordinary Internet users. The purpose of the project - to help both those who are not versed in cooking, but wants to eat correctly and delicious as those who simply need help, an advice or any opinion of other people.

«CookTasty» will please its simplicity, brightness, variety and completeness of the information.

Also in the project aspects of the proposed project economic efficiency, safety and environmental development, as well as its social significance are analyzed.

УДК 681.3.06

Палий Д.Н.

«Разработка веб-приложения для кулинаров “CookTasty”»

Дипломный проект –

ТТИ ЮФУ, 2013 г.- 80с.

РЕФЕРАТ

Дипломный проект содержит 80 страниц, 7 таблиц, 42 рисунка, библиографический список из 11 наименований.

В дипломном проекте рассмотрен подход к созданию веб-приложений. Для разработки изучены и рассмотрены различные технологии. В результате разработано функционирующее веб-приложение «CookTasty».

Также в проекте проанализированы аспекты экономической эффективности предложенных инженерных решений, безопасности и экологичности разработки, а также ее социальной значимости.

Оглавление

[ВВЕДЕНИЕ 6](#_Toc359084181)

[1 АНАЛИЗ ТЕХНИЧЕСКОГО ЗАДАНИЯ 8](#_Toc359084182)

[1.1 Постановка задачи 8](#_Toc359084183)

[1.1.1 Пользователи 8](#_Toc359084184)

[1.1.2 Роли 8](#_Toc359084185)

[1.1.3 Рецепты 9](#_Toc359084186)

[1.1.4 Комментирование 9](#_Toc359084187)

[1.1.5 Рейтинговая система 10](#_Toc359084188)

[1.1.6 Избранное 10](#_Toc359084189)

[1.1.7 Поиск 11](#_Toc359084190)

[1.1.8 Обратная связь 11](#_Toc359084191)

[1.2 Системные требования 11](#_Toc359084192)

[1.3 Требования к пользователю 11](#_Toc359084193)

[1.4 Вывод 11](#_Toc359084194)

[2 ОПИСАНИЕ РАЗРАБОТКИ 12](#_Toc359084195)

[2.1 Обзор средств и технологий разработки 12](#_Toc359084196)

[2.1.1 Обзор WAMP 12](#_Toc359084197)

[2.1.2 Другие использованные средства и инструменты 13](#_Toc359084198)

[2.1.3 Технология AJAX 14](#_Toc359084199)

[2.2 Требования к графическому интерфейсу 17](#_Toc359084200)

[2.2.1 Эскизы страниц 17](#_Toc359084201)

[2.2.2 Формат функциональных требований к графическому интерфейсу пользователя 23](#_Toc359084202)

[2.3 Построение архитектуры разрабатываемой системы 27](#_Toc359084203)

[2.3.1 Объектное моделирование 27](#_Toc359084204)

[2.3.1.1 Реализация разграничения прав пользователей 27](#_Toc359084205)

[2.3.1.2 Диаграмма вариантов использования 28](#_Toc359084206)

[2.3.2 Структура базы данных 28](#_Toc359084207)

[2.3.3 UML-диаграммы 29](#_Toc359084208)

[2.3.3.1 Выявление актеров 29](#_Toc359084209)

[2.3.3.2 Выявление вариантов использования 30](#_Toc359084210)

[2.3.3.3 Диаграмма последовательности 32](#_Toc359084211)

[2.3.3.4 Диаграмма кооперации 32](#_Toc359084212)

[2.3.3.5 Диаграмма схем состояний 33](#_Toc359084213)

[2.3.3.6 Назначение классов и их диаграмма 33](#_Toc359084214)

[3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 35](#_Toc359084215)

[3.1 Регистрация 35](#_Toc359084216)

[3.2 Вход под своим логином 35](#_Toc359084217)

[3.3 Добавление рецепта 36](#_Toc359084218)

[3.4 Страница рецепта 37](#_Toc359084219)

[3.5 Избранное 40](#_Toc359084220)

[3.6 ТОП 41](#_Toc359084221)

[3.7 Поиск 42](#_Toc359084222)

[3.8 Обратная связь 43](#_Toc359084223)

[3.9 Супер администратор 45](#_Toc359084224)

[3.10 Главная страница 46](#_Toc359084225)

[4 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ 49](#_Toc359084226)

[4.1 Обоснование необходимости и актуальности разработки 49](#_Toc359084227)

[4.2 Обоснование выбора аналога для сравнения 50](#_Toc359084228)

[4.3 Расчет затрат на проектирование и разработку 51](#_Toc359084229)

[4.4 Расчет цены 55](#_Toc359084230)

[4.5 Продвижение разработки 56](#_Toc359084231)

[4.6 Заключение 56](#_Toc359084232)

[5 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ РАБОТЫ 57](#_Toc359084233)

[5.1 Анализ безопасности 57](#_Toc359084234)

[5.1.1 Описание трудового процесса пользователя 57](#_Toc359084235)

[5.1.2 Анализ и оценка напряженности трудового процесса 57](#_Toc359084236)

[5.1.3 Дерево причин отказов 63](#_Toc359084237)

[5.1.4 Разработка защитных и профилактических мероприятий 64](#_Toc359084238)

[5.1.5 Разработка мероприятий, снижающих воздействие выявленных вредных факторов 66](#_Toc359084239)

[5.2 Анализ экологичности 69](#_Toc359084240)

[5.3 Вывод 72](#_Toc359084241)

[ЗАКЛЮЧЕНИЕ 73](#_Toc359084242)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 74](#_Toc359084243)

[ПРИЛОЖЕНИЕ А – ЛИСТИНГ ПРОГРАММЫ 75](#_Toc359084244)

[Recipe.php 75](#_Toc359084245)

# ВВЕДЕНИЕ

Каждый человек ежедневно сталкивается с определенными задачами, выполнять которые необходимо. Рассмотрим одну из таких задач. Это кулинария. Невозможно утверждать, что мы помним все рецепты наизусть, держим их в голове постоянно. Даже самым искусным поварам, прекрасным домохозяйкам, матерям и бабушкам время от времени требуется посмотреть состав какого-либо блюда или поочередность приготовления.

Веб-приложение «CookTasty» содержит в себе не только огромное количество рецептов разных кухонь мира, подразделение на категории, фотографии к рецептам, подробное описание приготовления, состав, но так же пользователь может увидеть на странице рецепта рейтинг положительных и отрицательных отзывов, подсчет голосов, прочитать и оставить комментарии после рецепта. Удобный поиск на сайте поможет быстро найти именно то, что нужно с подсветкой искомых слов.

Зарегистрированные пользователи могут добавлять свои рецепты в базу, заносить понравившиеся блюда в Избранное. Так же эти пользователи могут быть наделены правами администратора, что позволит им следить за жизнью и развитием сайта, вносить свои идеи и предложения, которые только приветствуются. Каждый зарегистрированный пользователь имеет возможность связаться с администратором при помощи вкладки Обратная связь. Эта форма очень проста, сообщение моментально прямо с сайта отправляется на почту администратора, что может исключить трату времени на посещение почтового ящика пользователя.

Считаю нужным представить еще один раздел сайта. Это ТОП – лист всех рецептов. Пользователю может стать интересно, что же людям нравится готовить, за какие рецепты они ставят положительные баллы. В этом случае стоит перейти на страницу с ТОП – листом. Там представлены 10 самых популярных рецептов, рейтинг которых имеет максимальное значение.

Главная страница «CookTasty» имеет вид блога, что выглядит очень привлекательно для пользователя.

Блог (англ. *blog*, от *we****b log*** — интернет-журнал событий, интернет-дневник, онлайн-дневник) — веб-сайт, основное содержимое которого — регулярно добавляемые записи, содержащие текст, изображения или мультимедиа. Для блогов характерны недлинные записи вре́менной значимости, упорядоченные в обратном хронологическом порядке (последняя запись сверху). Отличия блога от традиционного дневника обусловливаются средой: блоги обычно публичны и предполагают сторонних читателей.[1]

Каждый рецепт на главной странице представляет собой блок, в котором хранится изображение рецепта, краткое описание, дата добавления, автор рецепта, количество человек, которым понравился данный рецепт, и ссылка на переход непосредственно к его чтению.

На главной странице представлена пагинация, что визуально улучшит внешний вид сайта, а так же ускорит доступ к рецептам, если пользователь помнит примерную дату добавления.

1. АНАЛИЗ ТЕХНИЧЕСКОГО ЗАДАНИЯ
   1. Постановка задачи

Цель проекта – создание веб-приложения для кулинаров. Разрабатываемый проект должен позволять автоматизировать процесс ведения записи, просмотра, добавления, редактирования кулинарных рецептов, а так же предоставлять пользователям различные расширенные возможности.

* + 1. Пользователи

Пользователи являются одним из центральных понятий предметной области. Модель пользователя является основой для идентификации и аутентификации работающего с системой персонала, к которому относятся обычные администраторы, главный над администраторами и обычных пользователей, то есть людей, посещающих просто сайт и не обладающих специальными правами, а также для авторизации их в данном проекте.

* + 1. Роли

Роли пользователей определяются гибкой моделью определения прав доступа пользователей. Роли включают в себя набор привилегий, позволяющих разграничивать доступ к различным функциям системы.

Пользователям назначается роль, например обычный пользователь, который может быть зарегистрированным или нет.

У зарегистрированных пользователей больше прав на сайте. Они могут сами добавлять, редактировать или удалять рецепты, заносить в избранное. На странице данного рецепта указывается пользователь как автор рецепта. В превью рецепта на главной странице так же отображается, кто его добавил.

Зарегистрированные пользователи могут быть наделены правами администратора. Администраторы сайта несут в себе следующую смысловую нагрузку: имеют права на удаление или редактирование рецептов независимо от того, кто его опубликовал, то есть следят за состоянием базы рецептов. Так же к администраторам можно отнести пользователя, который обладает всеми правами обычного администратора, но есть еще один плюс ко всему. А именно назначать кого-либо из обычных пользователей, согласовав все, администраторами, то есть наделять правами, но так же и лишать этих прав. Это отражено на специальной странице, которая доступна только супер администратору, и реализовано доступно и просто выводом всех пользователей по блокам, где справой стороны от каждого находится иконка «плюс» или «минус», дать права или лишить их соответственно. Все это реализовано с помощью технологий «jquery», что позволяет не перезагружать страницу при каждом нажатии на иконки.

* + 1. Рецепты

Рецепт является одним из основных понятий в данном веб - приложении. Благодаря этой сущности возможно организовать вывод всей нужной информации из базы данных на сайт с разграничением доступа различным пользователям. Сам рецепт представляет собой набор большой информации. В ее состав входят название данного рецепта, дата добавления, автор, краткое содержание, которое отражается как на превью, так и на основной странице, кухня, к которой можно отнести рецепт, изображение, состав, с подробным описанием ингредиентов, их количеством и мерой измерения, подробное описание рецепта, занесенное в раздел «Приготовление».

* + 1. Комментирование

Раздел комментирования расположен на основной странице рецепта. Ниже полной информации по рецепту каждый пользователь может прочитать комментарии, может написать сам, независимо зашел он в систему или нет, что позволит оставлять записи, мнение, советы абсолютно любому пользователю. Это поможет составить более полное представление о рецепте.

* + 1. Рейтинговая система

Каждый рецепт имеет свой рейтинг. К рейтингу относятся как положительные, так и отрицательные отзывы. Реализовано это 2 кнопками – красной и зеленой. Красная кнопка означает, что рецепт не понравился, зеленая – обратное. Что на интуитивном уровне вполне понятно любому пользователю. Голосовать могут только зарегистрированные пользователи, причем один раз. При повторном щелчке по какой-либо кнопке пользователь получит сообщение, что с его учетной записи голосование уже произошло.

Рейтинг рецептов отображается так же в превью на главной странице. Это удобно тем, что пользователь сразу видит сколько голосов «за» и сколько «против» у данного рецепта, и по желанию может его не открывать из-за большого количества голосов «против» и наоборот.

Так же рейтинг используется еще в одном разделе сайта – «ТОП», где выводятся на рассмотрение только лучшие рецепты, которые собрали больше всего голосов «за». Эти рецепты имею вид превью как на главной странице.

* + 1. Избранное

Зарегистрированный пользователь может добавить любой рецепт себе в избранное. На панели слева появится ссылка «Избранное». При переходе на нее пользователь увидит список рецептов, которые он сам туда добавил. Рецепты из избранного можно удалить простым нажатием на красный крест в правом верхнем углу каждого блока с рецептом.

На сайте реализована технология резервного копирования. Если пользователь добавил в избранное рецепт, а потом автор или администратор его удалили по какой-либо причине, то в избранном рецепт все равно будет отображаться, но в поле «автор рецепта» высветится надпись «Было удалено». Это позволит пользователю обращаться к любимому рецепту, несмотря на его отсутствие в общем списке рецептов.

* + 1. Поиск

На сайте присутствует поиск по рецептам. Поиск по рецептам позволит пользователю за считаные секунды найти нужный ему рецепт либо по названию, либо по описанию. В итоге на странице поиска появится список всех рецептов в виде превью с подсветкой искомых слов. Поиск реализован без перегрузки страницы.

* + 1. Обратная связь

Обратная связь – раздел сайта, где любой пользователь, заполнив контактную форму, может отправить письмо администратору на почтовый ящик. После нажатия кнопки «Отправить сообщение» на месте формы появится запись, где будет отображено, отправлено ли данное сообщение успешно.

* 1. Системные требования

Конечный продукт может работать на ПК под любой ОС и платформой, имеющей веб-браузер.

* 1. Требования к пользователю

Пользователь и обычный администратор должны обладать навыками работы с ПК и владеть базовым функционалом интернет-браузера. Суперадминистратор системы, помимо этого, должен иметь навыки работы с СУБД и веб-сервером.

* 1. Вывод

В данном разделе сформулирована цель создания системы, требования к системе и пользователю. Определен будущий функционал системы.

1. ОПИСАНИЕ РАЗРАБОТКИ
   1. Обзор средств и технологий разработки
      1. Обзор WAMP

**WAMP** —  акроним, обозначающий набор (комплекс)  серверного программного обеспечения, широко используемый с целью разработки, предоставления веб-сервисов. WAMP назван по первым буквам входящих в его состав компонентов:

* [**W**indows](http://ru.wikipedia.org/wiki/Windows) — операционная система от компании [Microsoft](http://ru.wikipedia.org/wiki/Microsoft" \o "Microsoft);
* **A**pache — веб-сервер;
* **M**ySQL  — СУБД;
* **P**HP — язык программирования, используемый для создания веб-приложений.[2]

Основными достоинствами **Apache** считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации  пользователей, модифицировать сообщения об ошибках и т. д. [3]

**PHP** (англ.*PHP: Hypertext Preprocessor*  —«PHP: препроцессор гипертекста » ; первоначально *Personal Home Page Tools* — «Инструменты для создания персональных веб-страниц»; произносится *пи-эйч-пи*) — скриптовый язык программирования общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг -провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов.[4]

**MySQL**  —  [свободная](http://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%9F%D0%9E) [реляционная система управления базами данных](http://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94). Разработку и поддержку MySQL осуществляет корпорация [Oracle](http://ru.wikipedia.org/wiki/Oracle" \o "Oracle).

MySQL является решением для малых и средних приложений. Входит в состав серверов [WAMP](http://ru.wikipedia.org/wiki/WAMP), [AppServ](http://ru.wikipedia.org/w/index.php?title=AppServ&action=edit&redlink=1" \o "AppServ (страница отсутствует)), [LAMP](http://ru.wikipedia.org/wiki/LAMP) и в портативные сборки серверов [Денвер](http://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BD%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0)), [XAMPP](http://ru.wikipedia.org/wiki/XAMPP). Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.[5]

* + 1. Другие использованные средства и инструменты

Для обработки данных на клиентской стороне, визуального оформления, улучшения юзабилити и проверки данных, введенных пользователем, в системе используется Java Script, его библиотека jQuery, дополнение jQuery UI, а так же некоторые плагины.

Реализация одних и тех же функций в различных приложениях побуждает разработчиков заново писать один и тот же код несколько раз лишь незначительно изменяя его под конкретное приложение.

Плагины jQuery позволяют забыть разработчикам о данной проблеме. Разработчик может один раз написать плагин, который позволяет реализовать определенную функцию и затем использовать его в необходимых приложениях написав только одну строчку кода.

В интернете можно найти огромное количество бесплатных jQuery плагинов написанных другими разработчиками и использовать их для создания своих приложений.[6]

* jQuery AutoComplete – плагин, отвечающий за автозаполнение. Первое достоинство плагина — это производительность. Все результаты запросов кэшируются, и в следующий раз подбираются из кэша, а не с сервера.  
  Кроме того, у него есть несколько уникальных функций, такие как использование разделителей, поиск по нескольким словам, возможность использовать компонент автономно, без отправки запроса на сервер.
* jQuery Templates – это «движок шаблонов», работающий на стороне клиента как расширение jQuery. Этот плагин помогает показать в браузере данные, которые находятся в объектах и массивах JavaScript, избавляя от рутинных операций по созданию HTML-кода, экранированию специальных символов и т.п. Кроме того, он обладает очень интересными возможностями – например, позволяет обновлять созданный с его помощью HTML-код при изменении исходных данных.
* jQuery Validation – это плагин, позволяющий проверять данные форм. Проверка данных форм это чрезвычайно важная процедура для повышения «юзабилити» при работе с формами. В первую очередь потому, что пользователь сразу же получает уведомление в том случае, если какие либо данные введены неверно.
  + 1. Технология AJAX

**AJAX** — подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с [веб-сервером](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80). В результате, при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.

AJAX — не самостоятельная технология, а концепция использования нескольких смежных технологий. AJAX базируется на двух основных принципах:

* использование технологии динамического обращения к серверу «на лету», без перезагрузки всей страницы полностью, например:
  + с использованием [XMLHttpRequest](http://ru.wikipedia.org/wiki/XMLHttpRequest" \o "XMLHttpRequest) (основной объект);
  + через динамическое создание дочерних фреймов;
  + через динамическое создание тега <script>.
  + через динамическое создание тега <img>, как это реализовано в google analytics.
* использование DHTML для динамического изменения содержания страницы;

Действия с интерфейсом преобразуются в операции с элементами DOM (англ. *Document Object Model*), с помощью которых обрабатываются данные, доступные пользователю, в результате чего представление их изменяется. Здесь же производится обработка перемещений и щелчков мышью, а также нажатий клавиш. Каскадные таблицы стилей, или CSS (англ. *Cascading Style Sheets*), обеспечивают согласованный внешний вид элементов приложения и упрощают обращение к DOM-объектам. Объект XMLHttpRequest (или подобные механизмы) используется для асинхронного взаимодействия с сервером, обработки запросов пользователя и загрузки в процессе работы необходимых данных.[7]

На рисунке 2.1 можно увидеть модель классических приложений для сети (слева) в прямом сравнении с применением Ajax (справа).

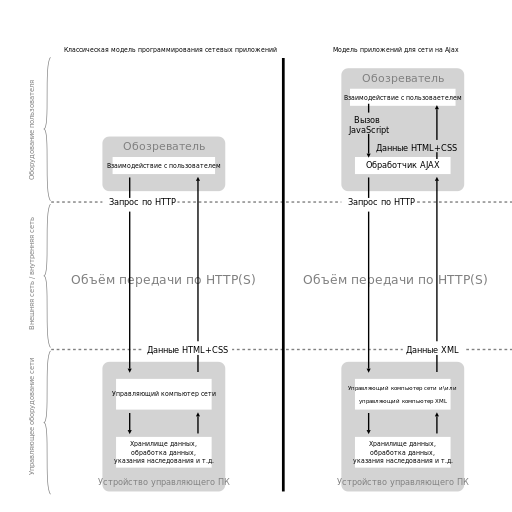


Рисунок 2.1 - Модель классических приложений для сети (слева) в прямом сравнении с применением Ajax (справа).

* 1. Требования к графическому интерфейсу

Этот раздел описывает требования к графическому интерфейсу пользователя, которые необходимо реализовать. Эскизы страниц приведены ниже.

* + 1. Эскизы страниц

*Имя/идентификатор страницы:* index.php

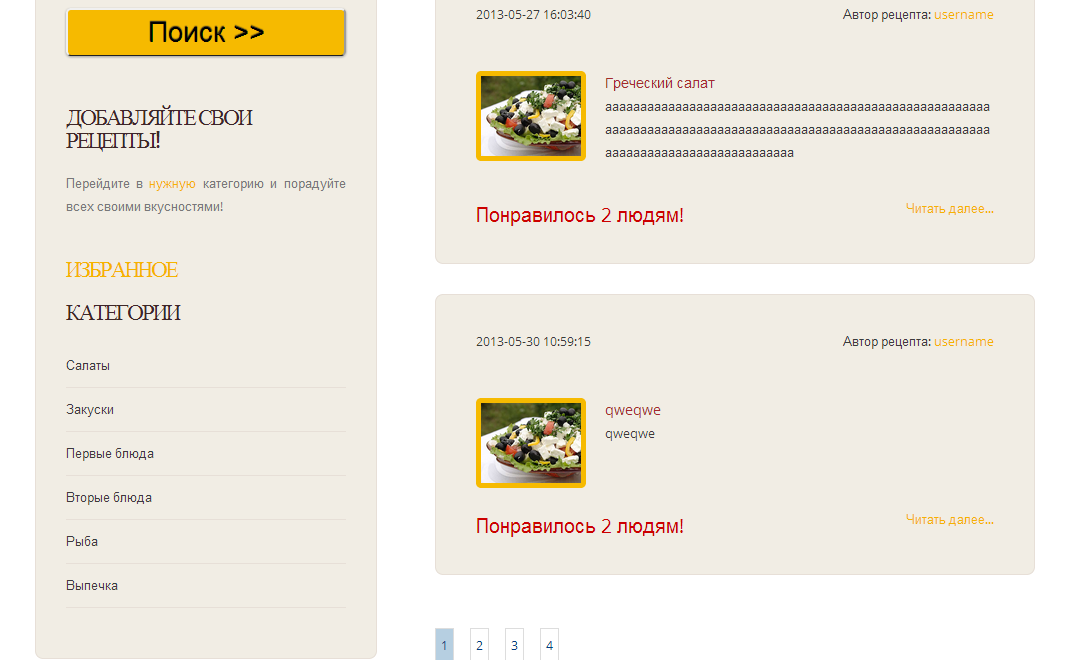


Рисунок 2.2 – страница «index.php»

*Имя/идентификатор страницы:* register.php

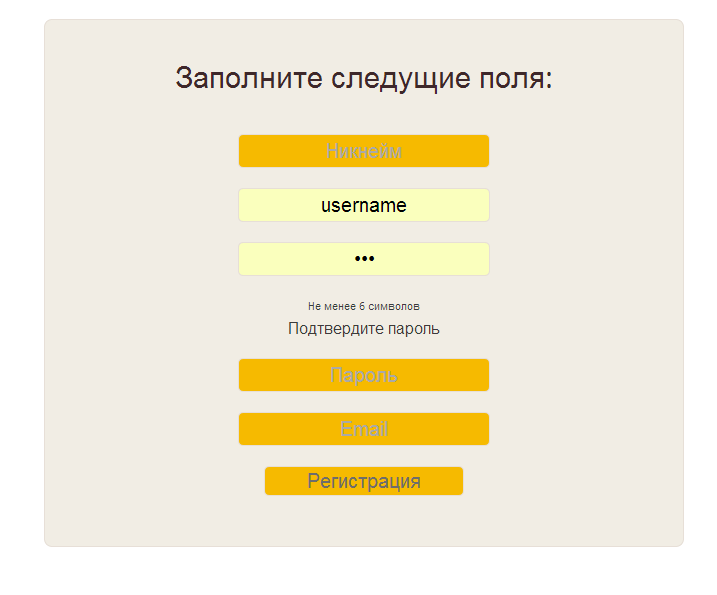


Рисунок 2.3 – страница «register.php»

*Имя/идентификатор страницы:* login.php

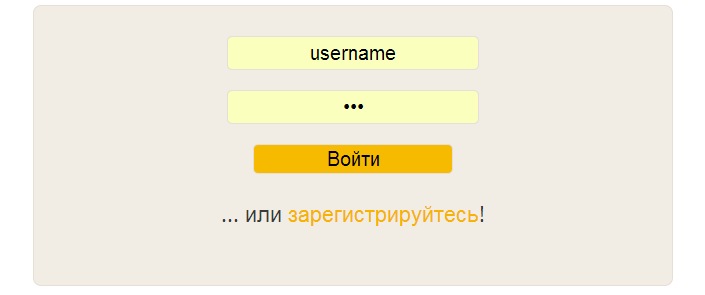


Рисунок 2.4 – страница «login.php»

*Имя/идентификатор страницы:* category.php

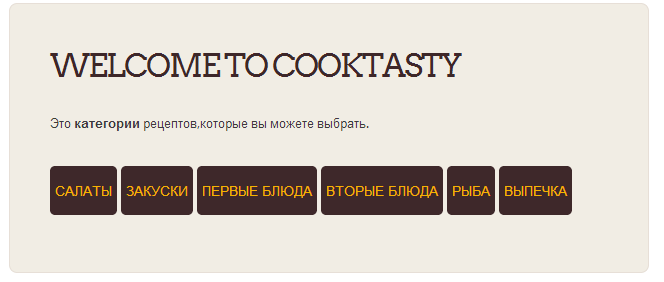


Рисунок 2.5 – страница «category.php»

*Имя/идентификатор страницы:* search.php

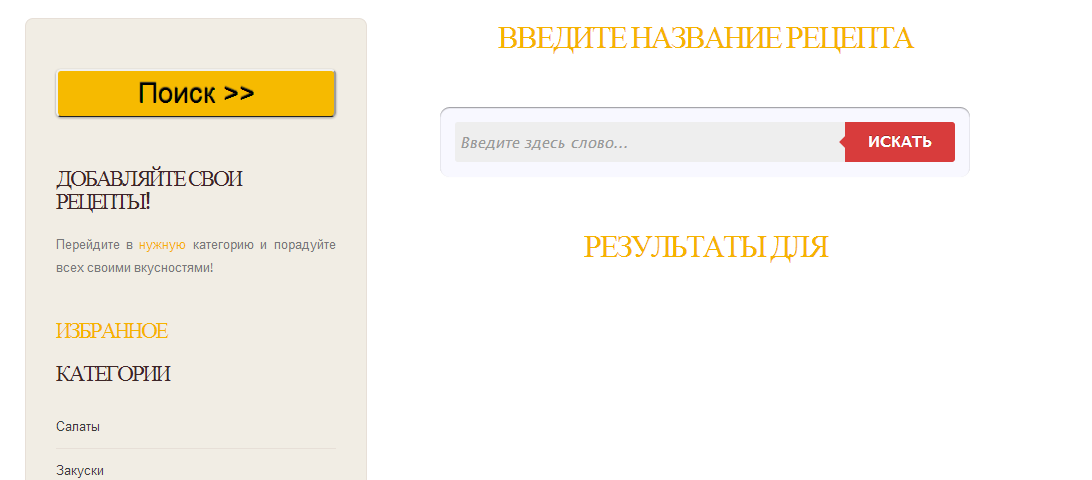


Рисунок 2.6 – страница «search.php»

*Имя/идентификатор страницы:* userFavorite.php

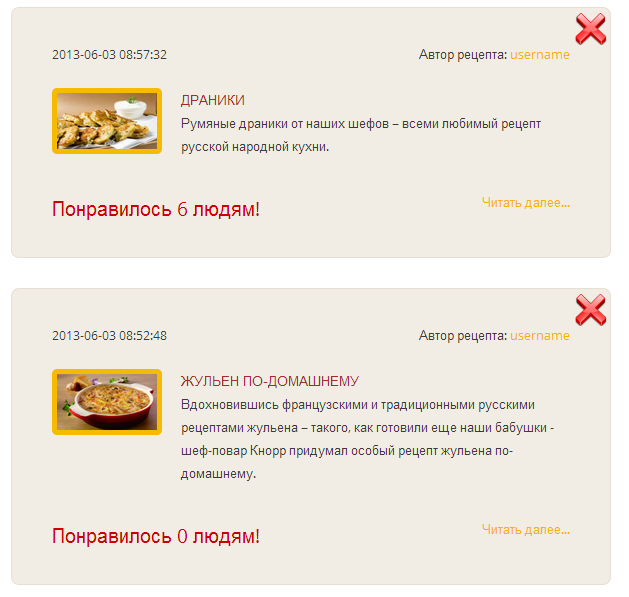


Рисунок 2.7 – страница «userFavorite.php»

*Имя/идентификатор страницы:* admin.php

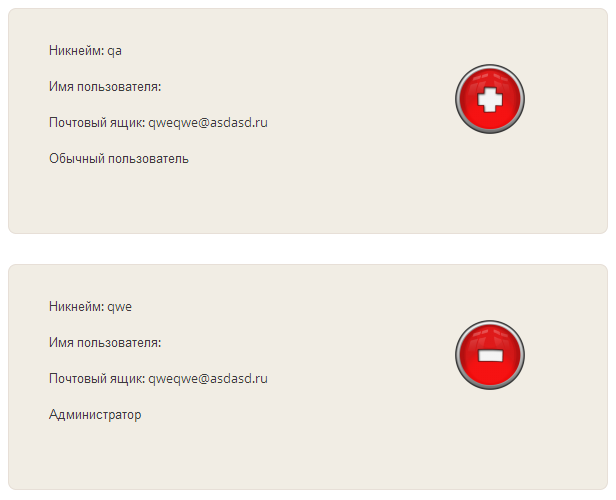


Рисунок 2.8 – страница «admin.php»

*Имя/идентификатор страницы:* meal.php



Рисунок 2.9 – страница «meal.php»

*Имя/идентификатор страницы:* add.php

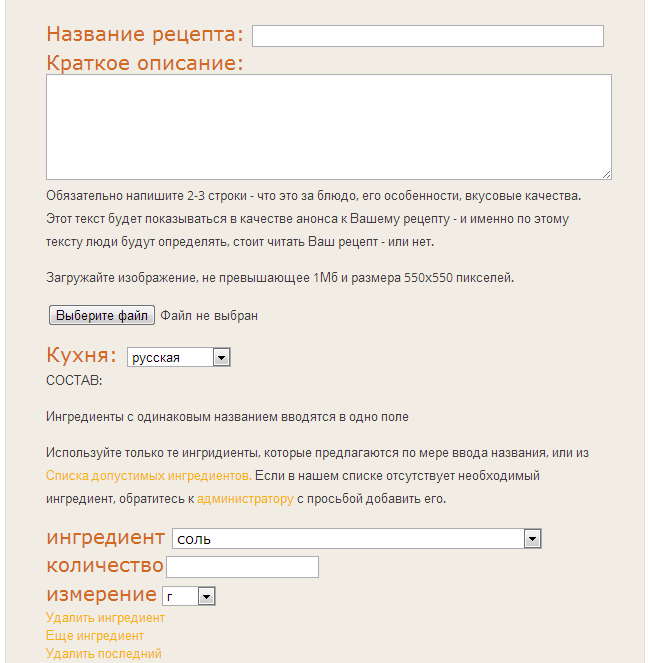


Рисунок 2.10 – страница «add.php»

*Имя/идентификатор страницы:* recipe.php

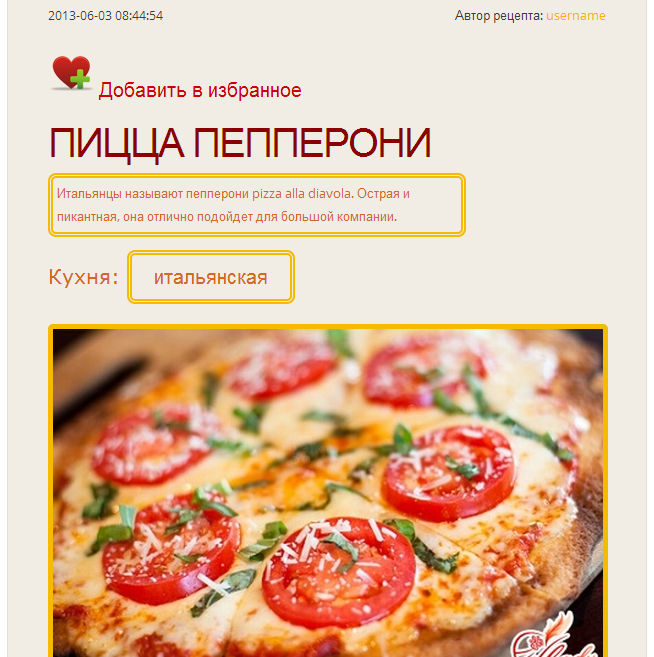


Рисунок 2.11 – страница «recipe.php»

*Имя/идентификатор страницы:* contact.php

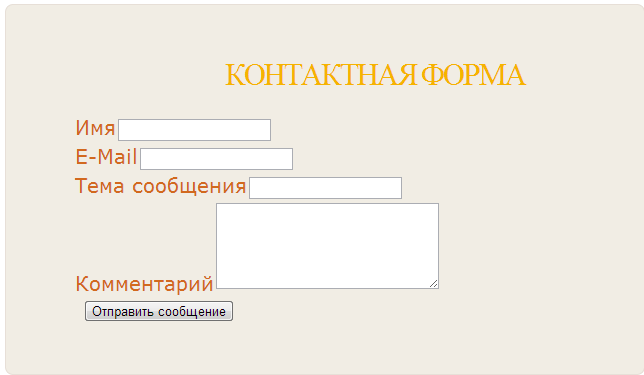


Рисунок 2.12 – страница «contact.php»

* + 1. Формат функциональных требований к графическому интерфейсу пользователя

Функциональные требования к графическому интерфейсу пользователя приведены в следующем формате.

*Имя:*

*Цель:*

*Тип элемента:*

*Индекс / идентификатор окна:*

*Действия:*

*Замечания:*

**Функции окон:**

1. index.php

*Имя:* Главная страница

*Цель:* Просмотр рецептов

*Тип элемента:* блоки со ссылками

*Индекс/ идентификатор:* index.php

*Действия:* Перейти на рецепт

*Замечания:* нет

1. register.php

*Имя:* Регистрация

*Цель:* Регистрация пользователя

*Тип элемента:* форма

*Индекс/ идентификатор:* register.php

*Действия:* Добавить данные о пользователе в БД через форму

*Замечания:* нет

1. login.php

*Имя:* Логирование

*Цель:* Авторизация пользователя

*Тип элемента:* форма

*Индекс/ идентификатор:* login.php

*Действия:* Проверить данные о пользователе в БД через форму

*Замечания:* нет

1. category.php

*Имя:* Категории рецептов

*Цель:* Выбрать категорию

*Тип элемента:* Ссылки

*Индекс/ идентификатор:* category.php

*Действия:* Перейти на страницу с категорией

*Замечания:* нет

1. search.php

*Имя:* Поиск

*Цель:* Найти рецепт

*Тип элемента:* форма

*Индекс/ идентификатор:* search.php

*Действия:* Вывести рецепты

*Замечания:* нет

1. userFavorite.php

*Имя:* Избранное

*Цель:* вывести избранные рецепты

*Тип элемента:* блоки со ссылками

*Индекс/ идентификатор:* userFavorite.php

*Действия:* Вывести рецепты

*Замечания:* нет

1. admin.php

*Имя:* Админ

*Цель:* вывести всех пользователей

*Тип элемента:* блоки со ссылками

*Индекс/ идентификатор:* admin.php

*Действия:* Давать и забирать права пользователей

*Замечания:* нет

1. meal.php

*Имя:* Блюда

*Цель:* вывести рецепты, соответствующие данной категории

*Тип элемента:* блоки со ссылками

*Индекс/ идентификатор:* meal.php

*Действия:* Вывести рецепты

*Замечания:* нет

1. add.php

*Имя:* Добавить

*Цель:* Добавить рецепт, соответствующий данной категории

*Тип элемента:* форма с кнопкой и ссылками

*Индекс/ идентификатор:* add.php

*Действия:* Добавить рецепт в БД

*Замечания:* нет

1. recipe.php

*Имя:* Рецепт

*Цель:* Вывести всю информацию по рецепту

*Тип элемента:* блоки

*Индекс/ идентификатор:* recipe.php

*Действия:* Просмотреть информацию по рецепту

*Замечания:* нет

1. contact.php

*Имя:* Обратная связь

*Цель:* Отправить сообщение администратору

*Тип элемента:* форма

*Индекс/ идентификатор:* contact.php

*Действия:* Отправить письмо

*Замечания:* нет

* 1. Построение архитектуры разрабатываемой системы
     1. Объектное моделирование

Процесс разработки программного обеспечения подразумевает создания множества артефактов, одним из видов которых являются модели. Модели – это абстракция, описывающая моделируемую систему с определенной точки зрения и на определенном уровне абстрагирования. Модели – это абстракции системы, которые создаются архитекторами и проектировщиками и представляют законченный взгляд на систему. Для создания моделей используются диаграммы UML.[8]

UML (Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования в основном программных систем. UML не является языком программирования, но в средствах выполнения UML-моделей как интерпретируемого кода возможна кодогенерация .[9]

* + - 1. Реализация разграничения прав пользователей

Относительно системы пользователи делятся на 4 группы:

1. Незарегистрированные пользователи. Не имеют прав и могут лишь просмотреть рецепты, оставить комментарий.
2. Зарегистрированные пользователи с привилегией «user». Пользователи. Обладают наименьшими правами среди зарегистрированных пользователей.
3. Зарегистрированные пользователи с правами «admin». Администраторы системы. Могут редактировать и удалять любые рецепты из базы данных.
4. Супер администратор с правами «root». Имеет возможность давать и забирать права у остальных администраторов.

Вся информация о пользователях, зарегистрированных в системе, хранится в таблице «users-info».

* + - 1. Диаграмма вариантов использования

В системе возможны следующие варианты использования:

* Регистрация нового пользователя;
* Добавление, редактирование, удаление рецептов
* Поиск рецептов
* Добавление рецептов в избранное
* Управление пользователями
* Рейтинговая система рецепта
* Комментирование рецепта
* Обратная связь
* ТОП рецептов

Диаграмма вариантов использования представлена в Приложении A. Она иллюстрирует, как можно использовать разрабатываемую систему, а также какие действующие лица имеют доступ к системе и их возможности. Вариант использования «Authorization» является абстрактным вариантом использования. В нашем случае он должен входить в состав практически всех конкретных вариантов использования (пользователь не может производить действий с системой, пока он не идентифицирован).

* + 1. Структура базы данных

На рисунке 2.13 представлена диаграмма БД, которая используется в веб-приложении.

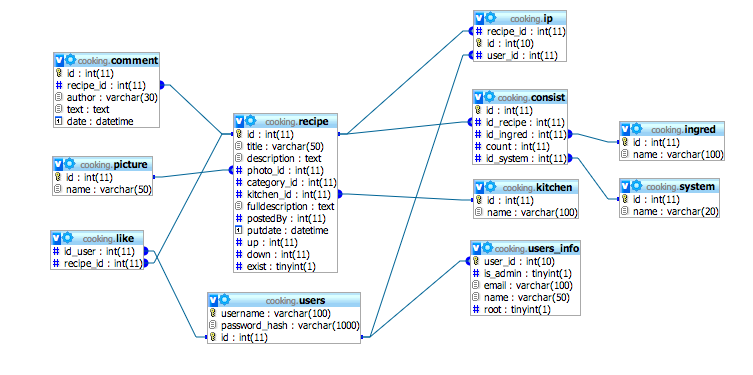


Рисунок 2.13 Структура базы данных

* + 1. UML-диаграммы
       1. Выявление актеров

На рис. 1 представлены основные кандидаты в акторы системы.

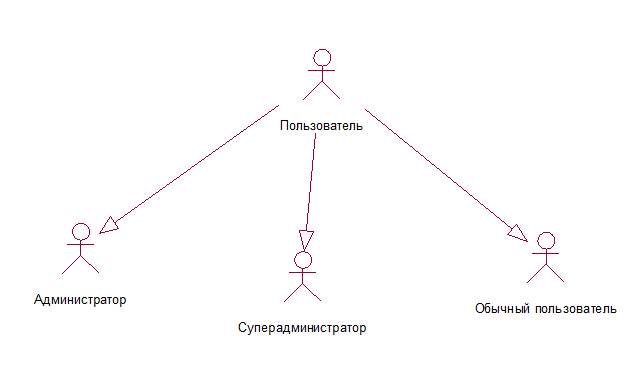


Рисунок 2.14 Анализ акторов системы

Краткое описание акторов представлено в табл. 1.

Табл. 1. Выявление акторов

|  |  |
| --- | --- |
| **Актёр** | **Краткое описание** |
| Администратор | Редактирует рецепты, удаляет и добавляет их. |
| Пользователь | Просматривает рецепты, добавляет, редактирует и удаляет только свои рецепты, добавляет в избранное рецепты, голосует за рецепт |
| Суперадминистратор | Дает и заирает права администратора обычным пользователям |

* + - 1. Выявление вариантов использования

Выявленные варианты использования представлены на рис. 2 и табл. 2.

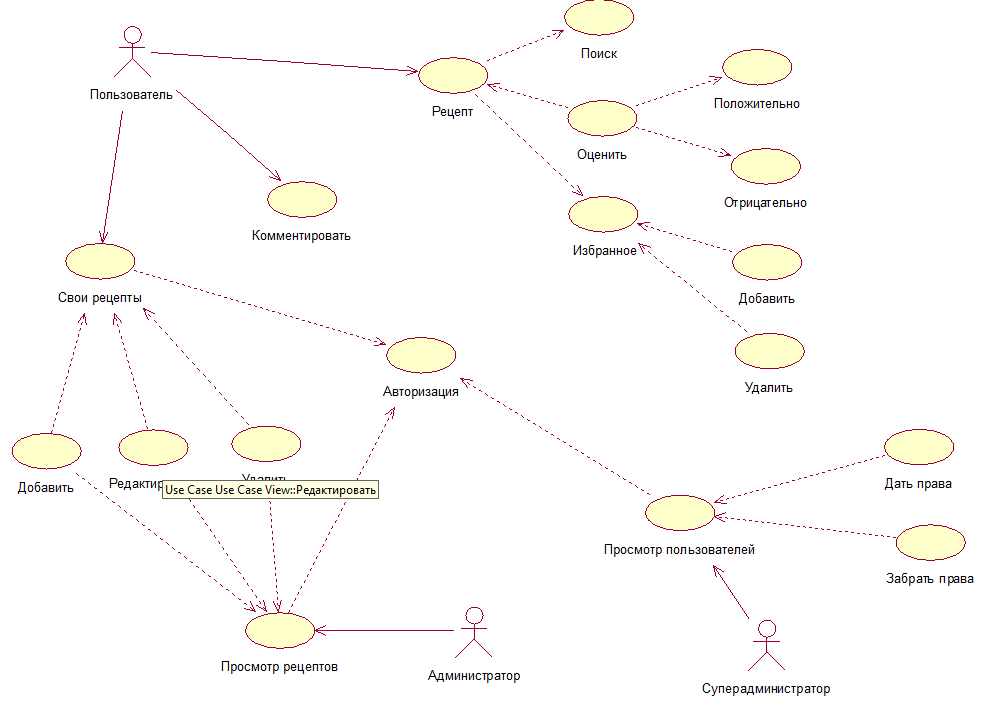


Рисунок 2.15 Диаграмма вариантов использования системы

Табл. 2. Краткое описание вариантов использования

|  |  |  |
| --- | --- | --- |
| Основной актёр | Наименование | Краткое описание |
| Администратор | Авторизация | Администратор вводит свое имя и пароль, после чего происходит его идентификация и разрешение прав. |
| Администратор | Просмотр рецептов | Добавление, удаление, редактирование рецептов. |
| Суперадминистратор | Просмотр пользователей | Администратор может давать и забирать права администратора. |
| Пользователь | Свои рецепты | Может добавлять, редактировать и удалять. |
| Пользователь | Комментировать | Комментирует свои и чужие рецепты |
| Пользователь | Рецепт | Пользователь может выполнять поиск и оценку рецептов, добавлять их в избранное. |
| Пользователь | Авторизация | Пользователь должен ввести свое имя и пароль для идентификации в БД |

* + - 1. Диаграмма последовательности

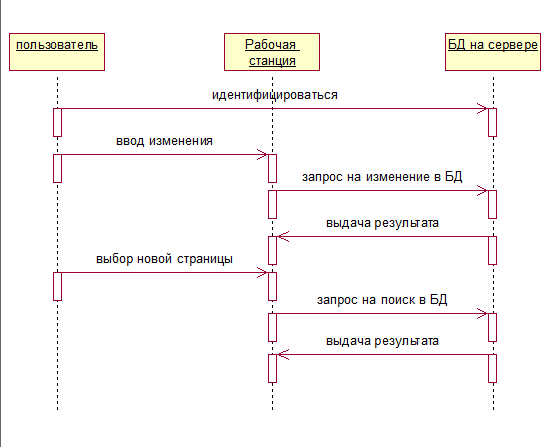


Рисунок 2.16 – Диаграмма последовательности

* + - 1. Диаграмма кооперации

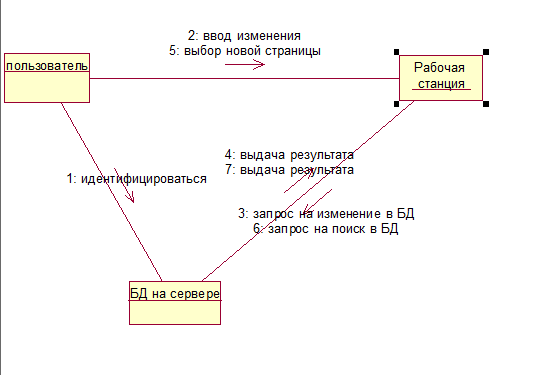


Рисунок 2.17 – Диаграмма кооперации

* + - 1. Диаграмма схем состояний

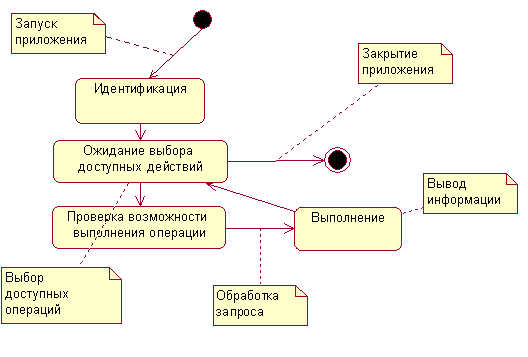


Рисунок 2.18 – Диаграмма схем состояний

* + - 1. Назначение классов и их диаграмма

Объектно-ориентированное или объектное программирование (в дальнейшем ООП) — парадигма программирования, в которой основными концепциями являются понятия объектов и классов.

Класс — это тип, описывающий устройство объектов. Понятие «класс» подразумевает некоторое поведение и способ представления. Понятие «объект» подразумевает нечто, что обладает определённым поведением и способом представления. Говорят, что объект — это экземпляр класса. Класс можно сравнить с чертежом, согласно которому создаются объекты. Обычно классы разрабатывают таким образом, чтобы их объекты соответствовали объектам предметной области.

Класс является описываемой на языке терминологии (пространства имён) исходного кода моделью ещё не существующей сущности, т. е. объекта.

Объект — сущность в адресном пространстве вычислительной системы, появляющаяся при создании экземпляра класса (например, после запуска результатов компиляции (и связывания) исходного кода на выполнение).

Прототип — это объект-образец, по образу и подобию которого создаются другие объекты.

На рисунке 2.19 представлена диаграмма основных классов ядра веб-приложения.

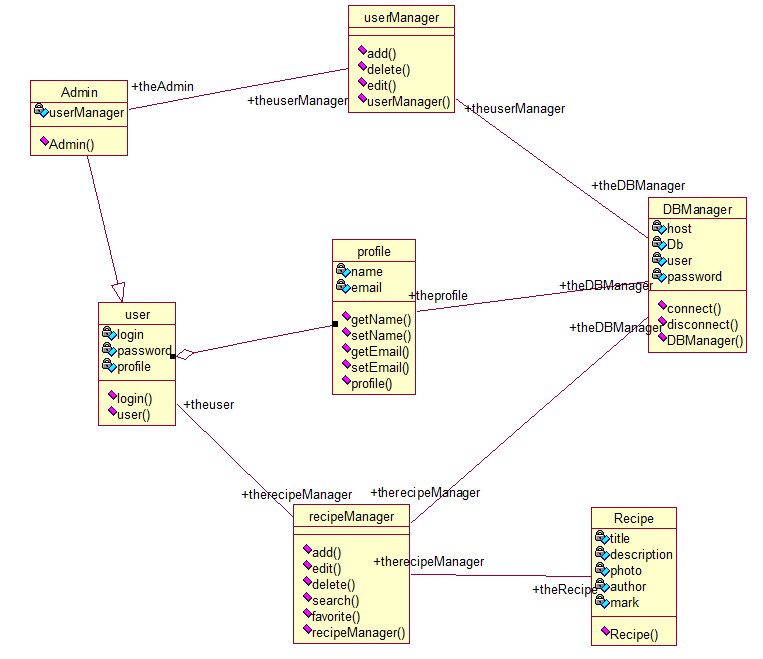


Рис. 2.19 – Диаграмма классов

Их взаимодействие и обеспечивает эффективную работу приложения.

1. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

В данном руководстве приведены основные действия при работе с «cookTasty» (далее сайтом) и их краткое описание.

* 1. Регистрация

Для того что бы выполнять большинство действий на сайте необходимо зарегистрироваться. Процесс регистрации максимально упрощён – необходимо ввести никнейм, имя пользователя, e‑mail и пароль (если пользователь с таким именем уже существует, выберите другое имя, которое будет отображаться на сайте). После успешной регистрации вы автоматически попадаете на главную страницу сайта.

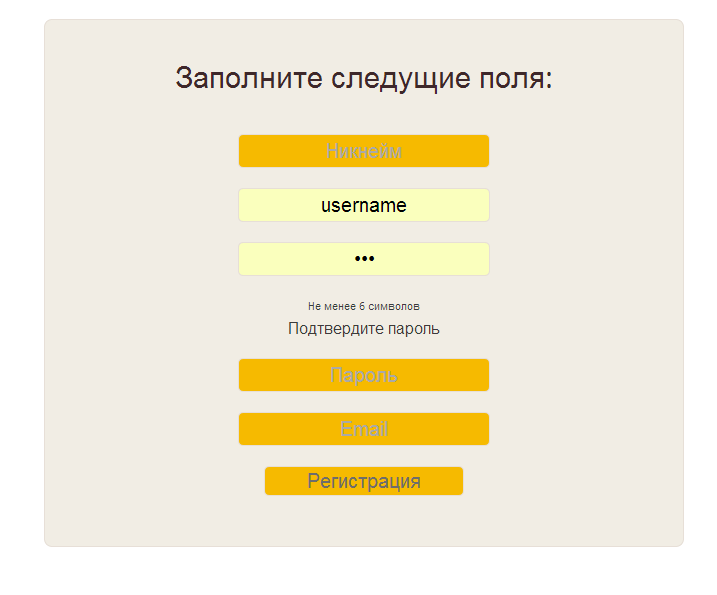


Рисунок 3.1 – Регистрация

* 1. Вход под своим логином

Для входа в систему необходимо быть зарегистрированным пользователем. Для регистрации в системе см. пункт 3.1.

Если пользователь зарегистрирован, то необходимо ввести логин и пароль в авторизационную форму. По завершении ввода необходимо нажать кнопку «Войти». Если данные были введены верно, то появится сообщение приветствия и кнопка выхода из системы.

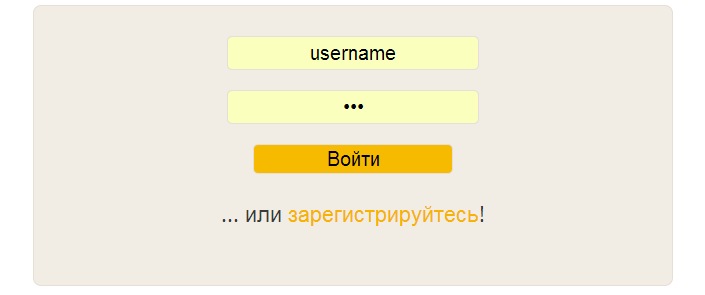


Рисунок 3.2 – Вход в систему

Если длина имени или пароля меньше 6 знаков или неправильно введены данные, появляется соответствующая запись справа от поля ввода.

* 1. Добавление рецепта

В соответствующей категории пользователь видит ссылку «Добавить новый рецепт».



Рисунок 3.3 – Добавить новый рецепт

Перейдя по ней, попадает на другую страницу, с формой для заполнения информации о рецепте.

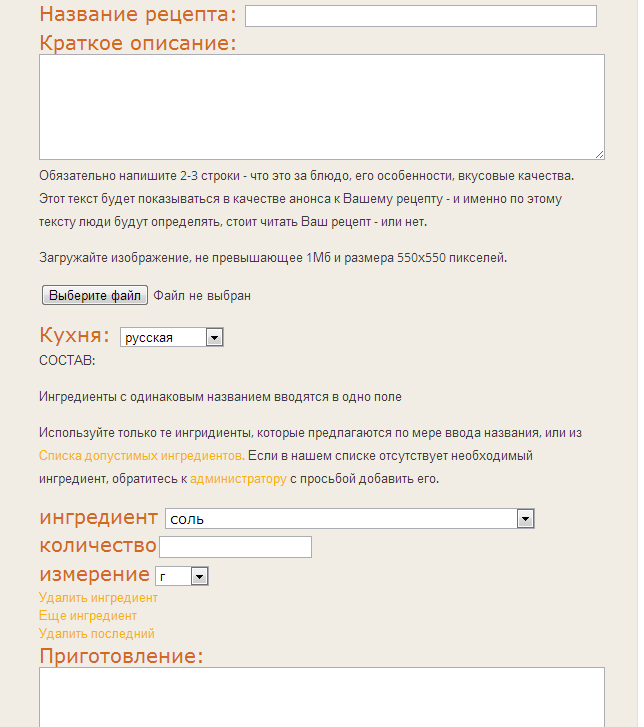


Рисунок 3.4 – Форма добавления рецепта

Заполнив все поля и нажав на кпонку «Добавить», пользователь добавляет тем самым рецепт в БД и попадает на страницу данного рецепта. Эту страницу рассмотри в следующем пункте 3.4.

* 1. Страница рецепта

Здесь содержится абсолютно вся информация по рецепту. Рассмотрим подробнее все возможности страницы.

В самом начале блока с рецептом находится дата добавления его в БД и автор, добавивший его. Ниже этого расположена иконка для добавления этого рецепта в Избранное. Каждый зарегистрированный пользователь может это сделать. По нажатию по иконке рецепт автоматически добавляется, а пользователь остается на той же самой странице.

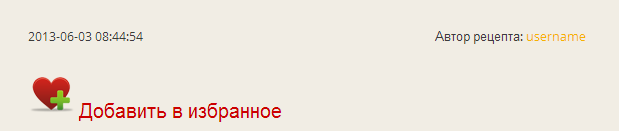


Рисунок 3.5 – Добавить в избранное

После этого идет вся информация по рецепту со всем подробным описанием и составом, а так же ссылки «Редактировать» и «Удалить» с левой стороны и оценка рецепта с правой.

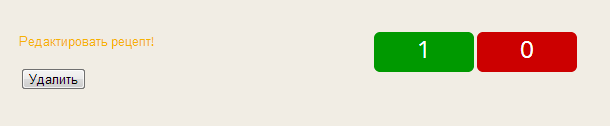


Рисунок 3.6 – Функционал после описания рецепта.

При нажатии на иконку зеленого цвета пользователь голосует ЗА рецепт, на красного – ПРОТИВ. В центре иконок выводится общее число голосов.

Посмотрим на следующий рисунок. (3.7)

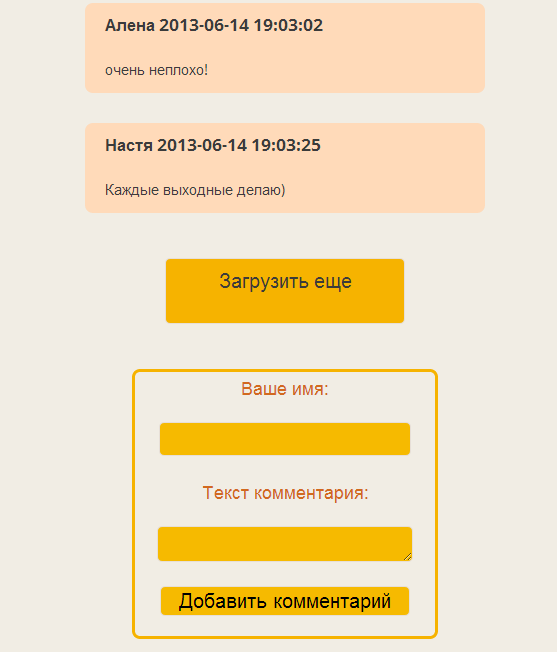


Рисунок 3.7 – Комментарии.

Итак, комментарии выводятся друг за другом. Нажав на кнопку «Загрузить еще», можно подгрузить больше комментариев с помощью технологии ajax. Ниже находится форма для заполнения комментария. При нажатии на кнопку «Добавить комментарий» комментарий тут же добавится, не перезагружая страницу.

Еще одна интересная часть данной страницы это скролл вверх.



Рисунок 3.8 – Скролл вверх.

При нажатии на него страница поднимется вверх до начала описания рецепта.

* 1. Избранное

Теперь рассмотрим «Избранное». Перейти на эту страницу можно при нажатии на ссылку на панели слева.



Рисунок 3.9 – Панель слева.

Затем пользователь может увидеть вывод всех рецептов, которые он добавил к себе.

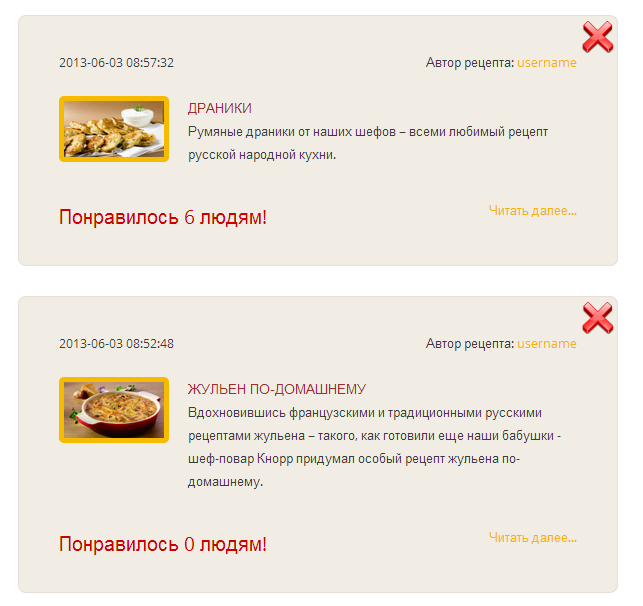


Рисунок 3.10 – Избранное.

Нажав на красный крестик вверху блока можно удалить из избранного рецепт.

Если рецепт был удален автором или администратором, в избранном он останется все равно, но с пометкой «Удалено».

* 1. ТОП

Если обратить внимание на рисунок 3.11, то можно увидеть такой раздел сайта как ТОП.



Рисунок 3.11 – ТОП.

Перейдя по этой ссылке, пользователь попадает на новую страницу, где отображены ТОП рецепты. Эти рецепты выбираются путем сортировки по голосам «ЗА», то есть по положительному голосованию. Пользователь может посмотреть, какие же рецепты самые популярные.

* 1. Поиск

Неотъемлемая часть практически любого сайта – это поиск. При переходе к поиску, появляется форма в правой части контента сайта. Форма очень удобна и проста.

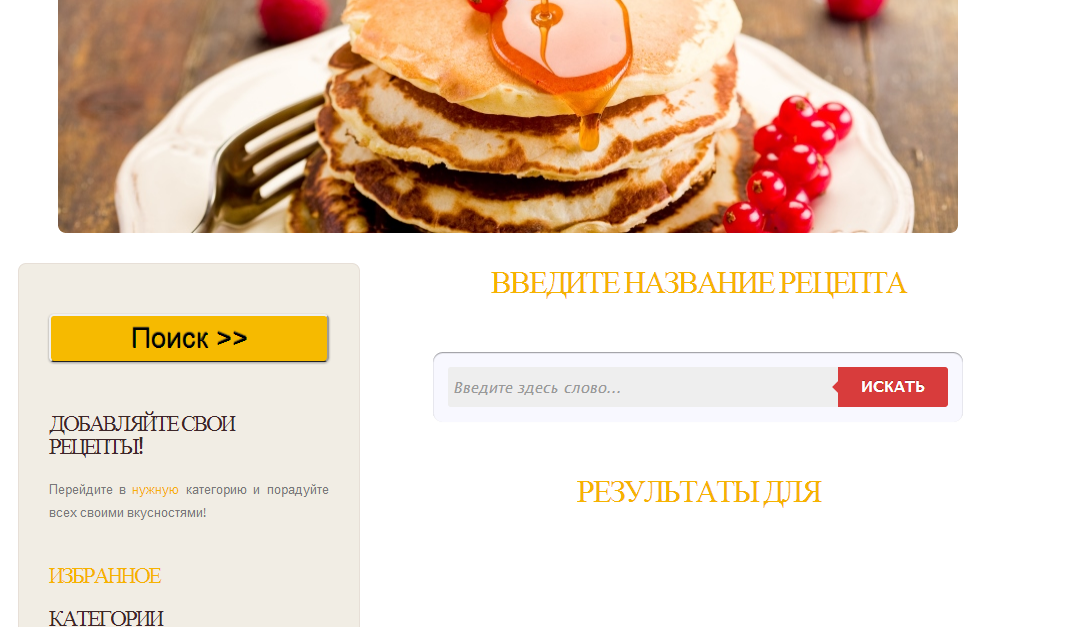


Рисунок 3.12 – Поиск.

Как только пользователь набрал искомое слово в поле и нажал на «Искать», без перегрузки страницы, благодаря «jquery», появляется все, что было возможно найти в базе данных по названию или описанию рецепта. Как представлено ниже на рисунке 3.13.

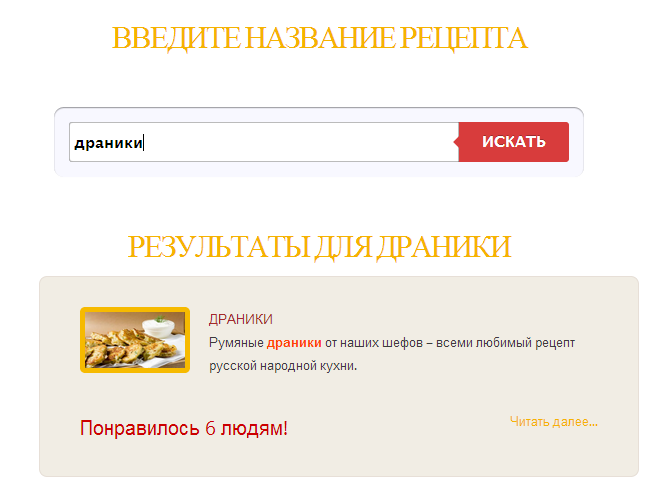


Рисунок 3.13 – Поиск.Результаты.

* 1. Обратная связь

Форма обратной связи уже давно является наиболее приемлемым способом общения между владельцем ресурса и пользователем.

Причин этому немало. В первую очередь, использование контактной формы в значительной мере облегчает отправку писем, благодаря специальным полям, отведенным для написания текста. Плюсом является то, что пользователям нет необходимости вводить адрес электронной почты владельца сайта, так как в форме обратной связи этот момент уже предусмотрен. Более того посетитель сайта даже не узнает электронного адреса администратора, что будет потенциальной защитой от спама.

 Помимо удобства для владельца ресурса и пользователя, форма обратной связи еще и существенно экономит время. Это происходит благодаря тому, что человеку, который хочет задать вопрос, не нужно тратить драгоценные минуты на запуск почтового клиента, чтобы отправить письмо. Все что требуется – это перейти на страницу с формой обратной связи, ввести в предназначенных для этого полях имя, адрес электронной почты, нужный текст и кнопку отправки.

Для того, чтобы создать форму обратной связи самостоятельно, необходимы специальные навыки, среди которых знание html, javascript и php.

Внешний вид формы очень прост (рисунок 3.14).

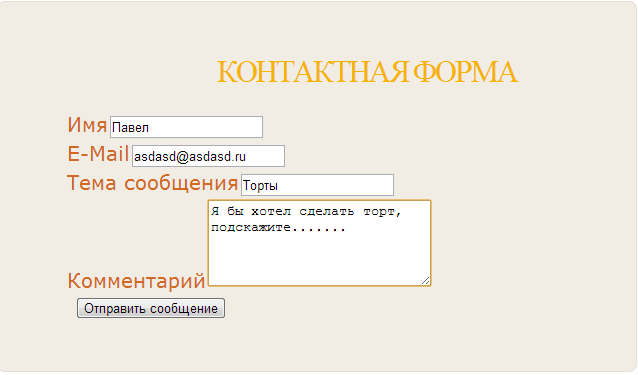


Рисунок 3.14 – Обратная связь.

После нажатия на кнопку «Отправить сообщение» вместо формы появляется результат отправки.

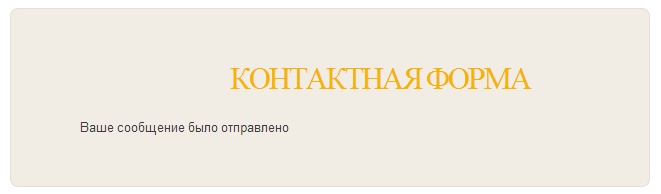


Рисунок 3.15 – Результат отправки сообщения.

* 1. Супер администратор

Это роль с максимальными возможностями. Супер администратор может управлять пользователями и видеть всю сеть. Он также известен как **Сетевой Администратор**, он ответственен за все добавления, редактирования и удаления рецептов, раздачу прав на администрирование пользователям, а так же и лишение этих прав, настройки сети и за сайт в общем. В обратной связи указан как раз адрес почтового ящика супер администратора.

На панели в заголовке сайта страница супер администратора обозначена иконкой, как показано на рисунке 3.16.



Рисунок 3.16 – Супер администратор.

Управление пользователями отображено ниже блоками с информацией по каждому пользователю (рисунок 3.17).

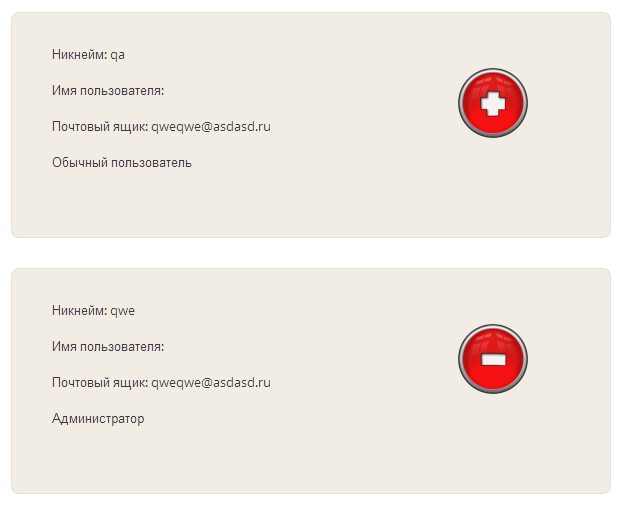


Рисунок 3.17 – Управление пользователями.

При нажатии на иконку «Плюс» супер администратор дает права администратора простому пользователю, на «Минус» - лишает их. Все это происходит при помощи обработки данных, которые передаются значению и id иконки, а так же отправке аякса, то есть все происходит моментально без перехода на другие страницы или перезагрузки всей страницы.

* 1. Главная страница

**Главная страница** – это информация, которая предстаёт перед пользователем при переходе его по адресу сайта. Другими словами, главная страница – это первое, с чем сталкивается посетитель, оказываясь на сайте. Правилу этому подчиняются все сайты в Интернете – контент-провайдеры, модные интернет-магазины, мощные порталы и многолюдные форумы. Предназначение главной страницы любого сайта – это обеспечение такого «приёма» посетителя, чтобы, в идеале, он стал пользователем. Или, по крайней мере, чтобы задержался на сайте в течение длительного времени.[10]

Главная страница, или по-другому **Домашняя,** состоит из нескольких частей. Рассмотрим их. Для начала обратим внимание на заголовок сайта или Header (рисунок 3.18).



Рисунок 3.18 – Заголовок сайта

Заголовок сайта состоит из панели в самом верху, где ссылками указаны главные разделы сайта, из названия сайта «CookTasty», приветствия текущего пользователя ниже и блока с изображениями различных блюд, которые переключаются с помощью «javascript jquery» через определенное время.

Слева расположена панель для быстрого доступа к определенным категориям рецептов, поиску и избранному.

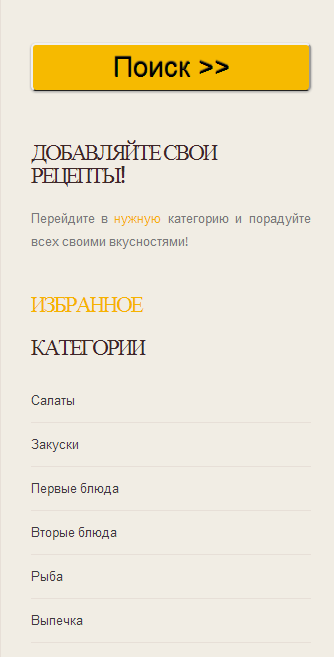


Рисунок 3.19 – Левая панель.

И наконец основная часть главной страницы (рисунок 3.20). Здесь в виде блога расположены друг за другом рецепты блоками в виде превью со ссылками «Читать далее…». Для удобства введена **пагинация страниц**.

Известно, что когда перед разработчиком стоит задача отобразить большое количество информации на сайте, он начинает думать о том, как сделать пагинацию (**нумерация страниц**): разбить контент на несколько страниц, где на каждой странице содержится какая-то часть информации. Обычно пагинация осуществляется со стороны сервера, то есть для каждой страницы мы извлекаем необходимое количество информации из базы данных. Но в случае, когда мы имеем дело с небольшим (средним) по величине контентом, такую пагинацию можно избежать: вместо нее можно использовать пагинацию со стороны пользователя, где работа со страницами осуществляется с помощью javascript. Что и представлено на данном сайте.



Рисунок 3.20 – Основная часть главной страницы.

1. ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ
   1. Обоснование необходимости и актуальности разработки

Целью разработки является создание веб – приложения для кулинаров. Разрабатываемое приложение должно позволять пользователю быстро, удобно и без лишних проблем и потери времени находить нужную ему информацию. Так же приложение должно позволять вести учет рецептов в базе данных. Сайт работает только при наличие сети Интернет, прекрасно поддерживается всеми браузерами.

Интерфейс приложения должен быть доступным и понятным в освоении как для разработчика, так и для пользователя.

В настоящее время разрабатывается всё больше и больше различных подобных веб - приложений и, как правило, этим занимаются целые компании, особенно если сайт привязан еще к какому-либо ресторану или кафе.

Кулинарные сайты очень популярны в наше время, и даже сложно представить, как бы люди без них обходились. Искать какие-то книги, вырезки из газет или журналов, идти к соседке или звонить родственникам и спрашивать какой-либо рецепт это долго, затратно и неудобно. Прекрасным решением проблем выступили сайты по кулинарии. От человека требуется всего лишь зайти на сайт и найти рецепт. Так же на сегодняшний день радует красота, яркость, удобство внешнего вида сайтов. Подробное описание приготовления блюда, плюс фотографии, отзывы, советы и комментарии других людей – все это поможет на кухне даже тому, кто готовит впервые.

Исходя из вышесказанного, становится ясна актуальность разработки, поскольку все сайты абсолютно разные, рецепты на них так же могут отличаться. И обычно что бы провести сравнение какого-либо рецепта, пользователь открывает ни один, а сразу 3-4 сайта с одинаковым блюдом, а потом просматривает, на каком из них ему больше нравится.

* 1. Обоснование выбора аналога для сравнения

Аналогов для сравнения можно привести очень много, так как данное направление очень развито.

Для примера будет взят сайт http://www.say7.info. Этот сайт является одним из самых популярных в сети Интернет.

Для наглядности его внешний вид продемонстрирован на рисунке 4.1.



Рисунок 4.1 - www.say7.info

Сведем все параметры рассмотренных аналогов и разработанного приложения в таблицу 4.1.

Таблица 4.1 – Обоснование выбора аналога для сравнения

|  |  |  |
| --- | --- | --- |
| Параметр | Разработаная система | Аналог |
| Наличие системы регистрации пользоватлей | да | нет |
| Разделение на категории рецептов | да | да |
| Лучшие рецепты | да | нет |
| Обратная связь | да | нет |
| Поиск по сайту | да | да |
| Добавление в избранное | да | нет |
| Добавление рецептов пользователями | да | нет |
| Комментирование рецептов | да | да |
| Использование технологий javascript | да | нет |
| Форум | нет | да |
| Новые рецепты | нет | да |
| Пагинация страниц | да | да |
| Наличие изображений блюд | да | да |
| Об авторе | нет | да |

* 1. Расчет затрат на проектирование и разработку

Расчет затрат на проектирование и разработку необходим для последующего расчета цены**.** Применительно к разработке программного обеспечения в себестоимость разрабатываемой системы включают следующее:

1. затраты на зарплату разработчиков;
2. затраты на эксплуатацию оборудования во время разработки программы;
3. прочие расходов (например, расходные материалы для принтера).

Все работы выполняются одним разработчиком – программистом.

Для расчета затрат на этапе проектирования определена продолжительность каждой из работ. Под проектированием понимается совокупность работ, которые необходимо выполнить, чтобы спроектировать систему или часть системы, или решить поставленную задачу. Данные о времени, затраченном на работу, сведём в таблицу 4.2.

Таблица 4.2 - Продолжительность работ на этапе проектирования и разработки

|  |  |  |
| --- | --- | --- |
| № | Наименование работ | Длительность работ (в часах) |
| 1 | Разработка ТЗ | 9 |
| 2 | Анализ ТЗ, работа с источниками | 24 |
| 3 | Разработка основных этапов работы | 4 |
| 4 | Общее проектирование модулей системы | 17 |
| 5 | Детальное проектирование модулей | 24 |
| 6 | Кодирование программы | 88 |
| 7 | Тестирование и отладка программы | 48 |
| 8 | Оформление пояснительной записки | 42 |
| 9 | Итого | 256 |

По данным таблицы 4.2 построим график организации работ на этапе проектирования и разработки в зависимости от времени (рисунок 4.2).

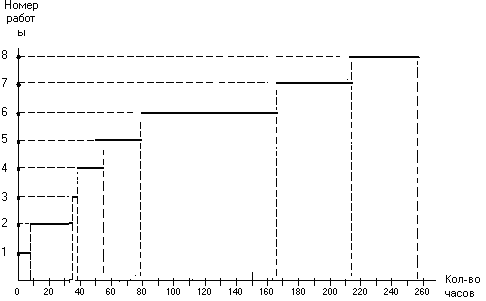


Рисунок 4.2 - График организации работ на этапе проектирования и разработки

Все работы выполнялись с использованием ЭВМ.

Поскольку в работе над проектом участвует только один разработчик, то параллельное выполнение тех или иных этапов невозможно. Время, которое планируется затратить на разработку программы, составляет Tпр = 256 часов.

Затраты на оплату труда разработчика при повременной форме (*Зпр*) рассчитываются исходя из его часовой ставки и времени его работы, формула (6).

, (1)

где *ас* – процент отчислений на социальное страхование, %;

*Тпр* – количество отработанных часов, ч;

*Сч* – почасовая оплата разработчика, руб.

Для расчета почасовой заработной платы примем, что средняя месячная заработанная плата разработчика программного обеспечения в городе Таганроге приблизительно равна средней заработной плате по Ростовской области размером в 13000 руб. Следовательно, часовая ставка:

*Сч = 13000 руб. / (25 \* 8) = 65 руб./ч*

С учетом этого затраты на оплату труда разработчика равны



Расходы на эксплуатацию оборудования вычисляются по формуле (2)

*Зэ = Зээ + А + Зпр****,*** (2)

где Зээ - затраты на электроэнергию,

А- затраты на использование оборудования (амортизация или аренда)***,***

Зпр - прочие затраты, связанные с эксплуатаций прибора (устройства).

Затраты на электроэнергию вычисляются по формуле (3)

*Зээ = Тмаш \* СкВт.час*, (3)

Тмаш - время эксплуатации ВТ для работ по проектированию и разработке прибора (устойства), по данным графика организации работ по проектированию,

СкВт час – тариф на электроэнергию.

Рассчитаем затраты на электроэнергию (*Зээ*), результаты расчета сведены в таблицу 4.3.

Таблица 4.3 – Затраты на электроэнергию

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование оборудования | Кол-во | Потребляемая мощность,  кВт | Часы работы | Тариф за  1 кВт\*ч., руб. | Стоимость электроэнергии, руб. |
| Блок питания ЭВМ | 1 | 0.07 | 256 | 3,08 | 55,2 |
| Итого | | | | | 55,2 |

Рассчитаем затраты на использование оборудования. Все работы выполнялись с использованием ЭВМ (ноутбук Sony VIAO), поэтому рассчитаем амортизационные расходы для персонального компьютера. Его стоимость составляет 30 тысяч рублей. Срок службы – 5 лет.

А = (30000 / (2002\*5)) \* 256 = 767.2 руб.

Для поиска нужной информации в интернете использовался безлимитный тариф 450 руб/мес. это значит что за каждый час проведенный в сети плата составляет 0,62 руб. Общее время проведенное в глобальной сети составило около 50 часов отсюда можно сделать вывод что затраты на интернет составили 31 рубль. Прочие затраты, связанные с эксплуатацией оборудования, отсутствуют.

Таким образом, затраты на эксплуатацию оборудования равны

Зэ = 55,2 руб. + 767.2 руб. + 31 = 853,4 руб.

В таблице 4.4 приведена калькуляция полной себестоимости системы.

Таблица 4.4 – Калькуляция полной себестоимости разработки

|  |  |
| --- | --- |
| Наименование статьи калькуляции | Сумма,руб. |
| Основная заработная плата | 22297,6 |
| Социальные отчисления (34 %) | 7581,2 |
| Расходы на содержание и эксплуатацию оборудования | 854 |
| Суммарные затраты на разработку | 30732,8 |

При вычислении полной себестоимости учтены основные статьи расходов.

* 1. Расчет цены

Рассчитаем продажную цену разработанного программного продукта, при условии, что планируемая прибыль от продажи должна составить не менее 30%. Такая прибыль обусловлена тем, что несмотря на то, что приложение рассчитано на широкий круг пользователей, на рынке уже существует хорошо себя зарекомендовавшие аналоги. В таблице 4.5 приводятся данные для определения цены продукции.

Таблица 4.5 – Расчет продажной цены

|  |  |
| --- | --- |
| Наименование статьи калькуляции | Сумма, руб. |
| Полная себестоимость | 30732,8 |
| Закладываемая прибыль (%) | 30 |
| Итого | 39952,6 |
| Цена с учетом НДС (18%) | 47144,1 |

Итак, цена реализации разработанного программного продукта составляет 47144 рубля.

* 1. Продвижение разработки

Разрабатываемая система носит популярный характер и будет интересна абсолютно любым пользователям сети Интернет, которые желают расположить данный ресурс в сети и возможно извлекать из него какую-либо выгоду.

Интересным синтезом данной области является объединение в один сайт и информации по рецептам и по ресторану или кафе, возможно прорекламировать какую-либо столовую с возможностью заказа у них готовых блюд.

Так же можно предположить, что доработка приложения в определенном направлении, например расчет меню на неделю или возможность считать калории и составлять низкокалорийные диеты, позволит ввести платные услуги и поднять прибыль от данного продукта.

* 1. Заключение

В результате всех проведенных расчетов были рассчитаны суммарные затраты на проектирование и разработку веб-приложения.

Преимущества над аналогом были явными, что приведено в таблице 4.1. Были рассмотрены возможные идеи продвижения сайта по направлению платных услуг и сервисов.

1. БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ РАБОТЫ
   1. Анализ безопасности
      1. Описание трудового процесса пользователя

Функционально представленный проект кратко отражен в разделе «Постановка задачи».

Работа с системой со стороны пользователя выглядит очень просто: ему предлагается интуитивно понятный интерфейс в виде страниц сайта с формами, элементами управления.

* + 1. Анализ и оценка напряженности трудового процесса

Напряженность труда оценивается по 22 факторам. По результатам проведенного анализа в таблице 5.1 представлен набор параметров, которые характеризуются наличием вредных производственных факторов, превышающих гигиенические нормативы и оказывающих неблагоприятное действие на организм работающего и/или его потомство.

Проанализируем и оценим напряженность трудового процесса, заполнив соответствующую таблицу 5.1 и 5.2.

Таблица 5.1 Анализ и оценка напряженности работы приложения для Администратора

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Интеллектуальные нагрузки: | | | |
| Наименование показателя | Заключение | Оценка | |
| 1 Содержание работы | Основные %работы связаны с администрированием системы и манипуляцией данными.  *Вывод* – решение простых задач по инструкции | 2.0 | |
| 2 Восприятие сигналов (информации) и их оценка | Восприятие сигналов с последующей коррекцией действий и операций | 2.0 | |
| 3 Распределение функций по степени сложности задания | Обработка, выполнение задания и проверка | 2.0 | |
| 4 Характер выполняемой работы | Работа по установленному графику | 2.0 | |
| 1. Сенсорные нагрузки: | | | |
| Наименование показателя | Заключение | Оценка | |
| 5 Длительность сосредото­ченного наблюдения (% времени смены) | 30 | 2.0 | |
| 6 Плотность сигналов (световых, звуковых) и сообщений в среднем за 1 ч работы | 65 | 1.0 | |
| 7 Число производственных объектов одновременного наблюдения | 3 | 1.0 | |
| 8 Размер %объекта различе­ния (при расстоянии от глаз работающего до объекта раз­личения не более 0,5 м) в мм при длительности сосредото­ченного наблюдения (% вре­мени смены) | 5-1,1 мм  более 50% | 2.0 | |
| 9 Работа с оптическими приборами (микроскопы, лу­пы и т. п.) при длительности сосредоточенного наблюдения (% времени %смены) | Нет | 1.0 | |
| 10 Наблюдение за экраном при работе:  с графической информацией  с цифровой информацией  (часов в смену) | До 5 | 2.0 | |
| 11 Нагрузка на слуховой анализатор | Разборчивость слов и сигналов от 100% до 90%. Помехи отсутствуют | 1.0 | |
| 12 Нагрузка на голосовой аппарат (суммарное количество часов, наговариваемое в неделю) | 12 | 1.0 | |
| 1. Эмоциональные нагрузки | | | |
| Наименование показателя | Заключение | | Оценка |
| 13 Степень ответственности за результат собственной деятельности. Значимость ошибки | Несет ответственность за выполнение отдельных элементов заданий. Ошибка влечет за собой дополнительные усилия в работе со стороны проектировщика | | 1.0 |
| 14 Степень риска для собст­венной жизни | Нет | | 1.0 |
| 15 Степень ответственности за безопасность других лиц | Нет | | 1.0 |
| 1. Монотонность нагрузок | | | |
| Наименование показателя | Заключение | | Оценка |
| 16 Число элементов (прие­мов), необходимых для реали­зации простого задания или в %многократно повторяющихся операциях | 7 | | 2.0 |
| 17 Продолжительность (в с) выполнения простых производственных заданий | Более 100 | | 2.0 |
| 18 Время активных действий (в % к продолжительности смены). %В остальное время – наблюдение за ходом производственного процесса | 20 и более | | 1.0 |
| 19 Монотонность производ­ственной обстановки (время пассивного наблюдения за хо­дом техпроцесса в % от време­ни смены) | Менее 75 | | 1.0 |
| 5. Режим работы: | | | |
| Наименование показателя | Заключение | | Оценка |
| 20 Фактическая продолжи­тельность рабочего дня | 6—7ч | | 1.0 |
| 21 Сменность работы | Односменная работа (без ночной смены) | | 1.0 |
| 22 Наличие регламентиро­ванных перерывов и их продолжительность | Перерывы регламентированы, доста­точной продолжительно­сти: 7 и%бо­лее рабочего времени | | 1.0 |
| Общая оценка напряженности |  | | 2.0 |

Так как 6 и более показателей отнесены ко 2 классу, а остальные - к 1 классу, то степень напряженности трудового процесса пользователя категории «Администратор» при работе с информационной системой может быть признана допустимой. Допустимые условия труда условно относят к безопасным, поэтому обязательным является соблюдение пользователем профилактических защитных мероприятий, регламентированных требований СанПиН 2.2.2/2.4.1340-03.

Таблица 5.2 - Анализ и оценка напряженности работы приложения для Клиента

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Интеллектуальные нагрузки: | | | |
| Наименование показателя | Заключение | Оценка | |
| 1 Содержание работы | Простые задачи по поиску информации в системе с интуитивно понятным интерфейсом.  *Вывод* – решение простых задач | 1.0 | |
| 2 Восприятие сигналов (информации) и их оценка | Восприятие сигналов с последующей коррекцией действий и операций | 1.0 | |
| 3 Распределение функций по степени сложности задания | Обработка, выполнение задания и проверка | 1.0 | |
| 4 Характер выполняемой работы | Основные %работы связаны с поиском информации по необходимости.  *Вывод* – работа выполняется по индивидуальному плану | 1.0 | |
| 1. Сенсорные нагрузки: | | | |
| Наименование показателя | Заключение | Оценка | |
| 5 Длительность сосредото­ченного наблюдения (% времени смены) | 30 | 2.0 | |
| 6 Плотность сигналов (световых, звуковых) и сообщений в среднем за 1 ч работы | 65 | 1.0 | |
| 7 Число производственных объектов одновременного наблюдения | 3 | 1.0 | |
| 8 Размер %объекта различе­ния (при расстоянии от глаз работающего до объекта раз­личения не более 0,5 м) в мм при длительности сосредото­ченного наблюдения (% вре­мени смены) | 5-1,1 мм  более 50% | 2.0 | |
| 9 Работа с оптическими приборами (микроскопы, лу­пы и т. п.) при длительности сосредоточенного наблюдения (% времени %смены) | Нет | 1.0 | |
| 10 Наблюдение за экраном при работе:  с графической информацией  с цифровой информацией  (часов в смену) | До 5 | 2.0 | |
| 11 Нагрузка на слуховой анализатор | Разборчивость слов и %сигналов от 100% до 90%. Помехи отсутствуют | 1.0 | |
| 12 Нагрузка на голосовой аппарат (суммарное количество часов, наговариваемое в неделю) | 12 | 1.0 | |
| 1. Эмоциональные нагрузки | | | |
| Наименование показателя | Заключение | | Оценка |
| 13 Степень ответственности за результат собственной деятельности. Значимость ошибки | В процессе эксплуатации не может изменить состояние системы, так как не обладает соответствующим набором прав доступа. Ошибка не влечет за собой дополнительные усилия в работе со стороны проектировщика | | 1.0 |
| 14 Степень риска для собст­венной жизни | Нет | | 1.0 |
| 15 Степень ответственности за безопасность других лиц | Нет | | 1.0 |
| 1. Монотонность нагрузок | | | |
| Наименование показателя | Заключение | | Оценка |
| 16 Число элементов (прие­мов), необходимых для реали­зации простого задания или в %многократно повторяющихся операциях | 7 | | 2.0 |
| 17 Продолжительность (в с) выполнения простых производственных заданий | Более 100 | | 2.0 |
| 18 Время активных действий (в % к продолжительности смены). %В остальное время – наблюдение за ходом производственного процесса | 20 и более | | 1.0 |
| 19 Монотонность производ­ственной обстановки (время пассивного наблюдения за хо­дом техпроцесса в % от време­ни смены) | Менее 75 | | 1.0 |
| 5. Режим работы: | | | |
| Наименование показателя | Заключение | | Оценка |
| 20 Фактическая продолжи­тельность рабочего дня | 6—7ч | | 1.0 |
| 21 Сменность работы | Односменная работа (без ночной %смены) | | 1.0 |
| 22 Наличие регламентиро­ванных перерывов и их продолжительность | Перерывы регламентированы, доста­точной продолжительно­сти: 7 и%бо­лее рабочего времени | | 1.0 |
| Общая оценка напряженности |  | | 2.0 |

В результате получили, что 17 и более показателей имеют оценку 1 класса, а остальные относятся ко 2 классу. При этом отсутствуют показатели, относящиеся к 3 (вредному) классу. Из этого следует, сто степень напряженности трудового процесса пользователя категории «Клиент» при работе с информационной системой может быть признана оптимальной. Оптимальные условия труда условно относят к безопасным, поэтому обязательным является соблюдение пользователем профилактических защитных мероприятий, регламентированных требований СанПиН 2.2.2/2.4.1340-03.

* + 1. Дерево причин отказов

На основании полученных данных построено дерево отказов (рисунок 5.1).



Рисунок 5.1 − Дерево отказов системы

* + 1. Разработка защитных и профилактических мероприятий

Учитывая все вышеперечисленное, можно порекомендовать следующие параметры организации рабочего места как разработчика программного продукта, так и пользователя:

- экран дисплея должен находится не ближе 76 см от лица человека-оператора (что соответствует длине вытянутой руки), так как на этом расстоянии интенсивность электромагнитного поля падает до безопасной величины менее 0.2 мкТ;

- необходимо, чтобы сзади и сбоку от дисплея не организовывались рабочие места на расстоянии менее 1.22 м.;

- при перерывах в работе с компьютером необходимо выключать дисплей.

Для профилактики возникновения синдрома длительных статических нагрузок желательно придерживаться следующих рекомендаций:

- при работе оператора с компьютером в течении длительного времени необходимо один раз в час на 15 мин прерывать работу с компьютером. Это время можно использовать для выполнения гимнастических упражнений;

- при появлении первых признаков СДСН: перенапряжений мышц, болезненных ощущений в запястьях, руках, плечах, шее или спине, онемений и покалываний, мышечных судорогах желательно обратиться за консультацией к врачу.

Для профилактики заболеваний глаз при работе с дисплеями желательно придерживаться следующих рекомендаций:

- дисплей не должен быть повернут экраном в сторону окна. В случае его расположения возле окна, необходимо расположить его перпендикулярно стеклу, для предотвращения возникновения бликов на экране;

- свет от осветительных ламп не должен падать на дисплей под углом более 60° от вертикали;

- освещенность рабочего места необходимо поддерживать в пределах двух третьих от нормальной освещенности. Это составляет порядка 300-500 лк;

- при освещении рабочего места неприемлемо использование мигающих источников света;

- интерьер, на фоне которого установлен дисплей, должен быть не ярким, не бросающимся в глаза. Блестящие предметы необходимо из него исключить, заменив их, по возможности на матовые. Освещенность интерьера должна быть примерно такая же, как и у дисплея. Соотношение яркости экрана и окружения не должно превышать 3:1;

- при работе желательно использовать специальные противобликовые фильтры, надеваемые на экран дисплея;

- правильный выбор видеоадаптера и монитора.

* + 1. Разработка мероприятий, снижающих воздействие выявленных вредных факторов

*Организационные*. В профилактике зрительного утомления и перенапряжения особую значимость имеет рациональный режим труда и отдыха, включающий регламентацию продолжительности рабочего дня, введение регламентированных перерывов, сеансов релаксации, выполнение специальных упражнений для глаз, соблюдение рекомендаций по организации активного отдыха.

Рабочее место должно обеспечивать человеку физиологически рациональную рабочую позу, соответствующую характеру и условиям труда. Рабочее положение должно соответствовать критериям комфорта и характеризуется: выпрямленным положением позвоночного столба, минимальной нагрузкой на мышечную систему тела человека, отсутствием болезненных ощущений.

*Технические мероприятия*. Наиболее радикальными средствами улучшения условий труда в условиях зрительного и умственного напряжения являются совершенствование техники и технологии, внедрение устройств для считывания и восприятия зрительной информации, отвечающих современным требованиям эргономики, использование рационального производственного освещения, цветовых контрастов. Благоприятное влияние на зрение оказывают относительно малонасыщенные цвета средней части видимого спектра (желтый, зеленый, голубой). В разработанной веб-системе простой и интуитивно-понятный пользователю интерфейс, отсутствуют резкие контрастные цвета, что в совокупности уменьшает зрительную утомляемость при работе с сайтом. Также для снижения влияния фактора зрительной утомляемости и исключения воздействия электромагнитного излучения от монитора, является целесообразным использование в лаборатории TFT мониторов (дисплеи на жидких кристаллах), вполне отвечающие требованиям эргономичности и безопасности, само устройство которых уже исключает наличие фактора вредного воздействия электромагнитного излучения. Рекомендуется правильно подбирать монитор, ориентируясь на современные стандарты защиты MPR-II, ТСО-99, TCO-03.

Для обеспечения электробезопасности при пользовании системой рекомендуется:

* применение малых напряжений;
* электрическое разделение сетей;
* контроль и профилактика повреждений изоляции;
* обеспечение недоступности токоведущих частей;
* применение защитного заземления;
* использование двойной изоляции;
* защитное (аварийное) отключение приборов.

*Индивидуальные*. Для уменьшения статическо-динамической физической нагрузки необходимо после каждого часа работы проводить производственную гимнастику (5-7 мин.). Длительная сидячая работа с фиксированным (особенно с наклоном в одну сторону) положением головы приводит к ухудшению мозгового кровообращения. Упражнения с наклонами и поворотами головы повышают эластичность кровеносных сосудов, питающих мозг, вызывают их расширение. Все это вместе с ритмичным дыханием через нос увеличивает приток кислорода к клеткам мозга, улучшает умственную работоспособность. Для снижения нервно-эмоциональной нагрузки ввести частые 3-4 мин. перерывы так чтобы время отдыха составляло 25% от общего времени работы за терминалом.

Также для уменьшения утомляемости зрительного аппарата необходимо:

* сохранять соотношение освещенности экрана и документа как 1/10;
* сохранять яркость фона меньше чем документа;
* использовать рассеянный свет;
* экран дисплея располагать ниже уровня глаз с наклоном;
* сохранять оптимальное расстояние от глаз до экрана (около 80 см).

Таким образом, на основании проведенного анализа можно сделать вывод, что для обеспечения наиболее удобной и безопасной работы необходимо принимать следующие меры:

*Организационные*: обучение и профессиональный отбор персонала, организация оптимальных (с точки зрения ресурсосбережения, производительности труда, безопасности) режимов работы оборудования и персонала, а так же контроль состояния производственной среды, эксплуатируемого оборудования и т.д.

*Технические:* необходим ввод в эксплуатацию современного оборудования, отвечающего последним стандартам безопасности и экологичности, применение передовых технологий и средств экобиозащитной техник.

*Санитарно-гигиенические:* обеспечение соответствующих условий эксплуатации оборудования, а так же обеспечение нормальной работы персонала, соответствующей требованиям действующих нормативных правовых актов (государственных стандартов, санитарных правил, норм, гигиенических нормативов).

*Психологические:* выявление истинных причин некорректных, ошибочных действий персонала(администраторов), совершенствование параметров и возможностей человеко-машинных интерфейсов, более полный учет основных особенностей личности и состояния человека в условиях напряженной производственной деятельности.

* 1. Анализ экологичности

При разработке программного продукта использовались экологически чистые технологии: информационные и компьютерные. Однако и в этом случае, существует косвенная вероятность загрязнения окружающей среды. Здесь основными вопросами экологичности является рациональное природопользование и ресурсосбережение.

Вопросы ресурсосбережения относятся как к материальным ресурсам, так и к энергетическим. Используемый в ходе работы персональный компьютер (ПК) потребляет электричество, при выработке которого наносится вред окружающей среде. Поэтому необходимо свести потери энергии к минимуму, используя энергосберегающие технологии. Кроме того, можно перевести ПК в дежурный энергосберегающий режим (или отключить). При возникновении задачи, требующей решения, он переводится в рабочее состояние. После решения задачи осуществляется обратный переход в дежурный режим. Кроме этого энергия потребляется на освещение рабочего места. Здесь можно использовать энергосберегающие лампы, а также вести работу, преимущественно ограничиваясь дневным светом.

Итак, экологичность проекта рассматривается с точки зрения экономии ресурсов. Организация сбора, преобразование, хранение и изменение данных стала самостоятельной задачей, от эффективности решения которой существенно зависит эффективность программного продукта. Чем быстрее работает программный продукт, тем меньше тратится энергии.

Помимо затраты ресурсов, компьютерные технологии сами по себе наносят вред окружающей среде. ПК является источником электромагнитных импульсов. Основными составляющими частями персонального компьютера являются: системный блок (процессор) и разнообразные устройства ввода/вывода информации. ПК часто оснащают сетевыми фильтрами, источниками бесперебойного питания и другим вспомогательным электрооборудованием. Все эти элементы при работе ПК формируют сложную электромагнитную обстановку на рабочем месте пользователя. Для снижения этого воздействия необходимо производить *организационные меры защиты,* направленные на обеспечение оптимальных вариантов расположения объектов, являющихся источниками излучения, и объектов, оказывающихся в зоне воздействия, организацию труда и отдыха персонала.

Одним из компонентов загрязнения окружающей среды является ионизация воздуха лазерными принтерами при печати и CD-ROM при записи дисков. В первом случае можно осуществлять печать на струйных принтерах. Во втором случае – можно использовать другие накопители информации.

С появлением таких информационных систем уменьшается потребление бумажных документов, что в свою очередь приводит к уменьшению производства бумаги, и как следствие вырубки леса, загрязнению окружающей среды при его переработке. Так же использование разработанной подсистемы позволяет существенно экономить время. Сэкономленное время можно потратить на решение других важных задач, что в свою очередь опять приведет к экономии денежных и природных ресурсов, а сэкономленные средства могут быть направлены на экологические цели.

При разборке и переработке ЖК дисплеев наибольшую угрозу окружающей среде представляют ртутные лампы. Процесс утилизации значительно упрощается, когда лампа может быть легко демонтирована и обработана отдельно. Лампа должна легко отсоединяться без риска быть повреждённой. Поэтому недопустимо приклеивание или приваривание (пайка) соединительных элементов. Количество ламп и ртути в них обязательно декларируется. Для каждой лампы указываются: поставщик (изготовитель), идентификатор изделия, среднее, максимальное и минимальное содержание ртути (в миллиграммах).

Существует несколько параметров, значения которых необходимо контролировать для снижения негативного воздействия компьютера на человека. К ним относятся уровни электромагнитных полей, акустического шума, мягкое рентгеновское излучение.

Согласно СанПиН 2.2.2/2.4.1340-03 временно допустимое значением напряженности электрического поля, создаваемого компьютером, должно находиться в диапазоне 2,5 В/м – 25 В/м, а плотность магнитного потока – в диапазоне 25 нТл – 250 нТл в зависимости от значения частоты.

Кроме этого, согласно СанПиН 2.2.2/2.4.1340-03 среди требований, которым должен удовлетворять компьютер, имеется нормативное значение для мощности экспозиционной дозы мягкого рентгеновского излучения в любой точке на расстоянии 0,05 м от экрана (на электроннолучевой трубке) при любых положениях регулировочных устройств. Оно составляет 1 мкЗв/час (100 мкР/час).

Для уменьшения воздействия электромагнитного и рентгеновского излучения рекомендуется использовать жидкокристаллические мониторы.

Уровень шума на рабочих местах при использовании компьютера так же имеет установленные нормы. При этом устройства, уровни шума которых превышают допустимые значения, по возможности должны выноситься в отдельные помещения. В противном случае необходимо использовать малошумные устройства.

Так же надо учесть то, что компьютер при работе использует достаточно большое количество электроэнергии. Поэтому, согласно ГОСТ Р51387-99, для уменьшения энергопотребления необходимо использовать специальные режимы работы компьютера.

Необходимо помнить о вреде экологии при утилизации ЭВМ.

Для предотвращения загрязнения окружающей среды утилизацию ЭВМ путем сжигания необходимо производить с использованием специальных полигонов, построенных в соответствии с требованиями СНиП 2.01.28-85.[11]

* 1. Вывод

В данном разделе проанализированы вопросы безопасности и экологичности при разработке системы. Проведен анализ напряженности трудового процесса для различных типов пользователей и установлено, что степень напряженности трудового процесса пользователей при работе с информационной системой может быть признана допустимой. Построено дерево причин отказов. Проведен анализ экологичности проекта. Определены меры профилактики и повышения безопасности и экологичности разрабатываемой системы.

# ЗАКЛЮЧЕНИЕ

Целью работы являлась разработка веб-приложения для кулинаров «cookTasty».

Для реализации поставленной задачи дипломного проекта был подробно проработан и систематизирован вопрос основ создания веб-приложения. Был составлен список требований к системе и проведён сравнительный анализ с существующими аналогичными веб-приложениями.

При экономическом анализе были рассчитаны показатели сравнительной технико-экономической эффективности, на основании которых можно сделать вывод, что разработка и внедрение данного продукта являются целесообразными.

Был произведен анализ вредных факторов, воздействующих на человека в процессе разработки и эксплуатации данного программного продукта, а так же указаны мероприятия по их снижению и профилактике.

Анализ безопасности и экологичности эксплуатации приложения показал, что степень напряжённости трудового процесса при работе пользователя с ним может быть признана допустимой, а сам продукт, при соблюдении необходимых профилактических защитных мероприятий, является безопасным.

Поставленная задача дипломного проектирования выполнена в полном объеме.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Блог [Электронный ресурс]. Дата обращения:5.06.2013. /<http://ru.wikipedia.org/wiki/%D0%91%D0%BB%D0%BE%D0%B3>.
2. Apache [Электронный ресурс]. Дата обращения:5.06.2013. <http://ru.wikipedia.org/wiki/Apache>
3. WAMP [Электронный ресурс]. Дата обращения:5.06.2013. <http://ru.wikipedia.org/wiki/Wamp>
4. PHP [Электронный ресурс]. Дата обращения:5.06.2013. <http://ru.wikipedia.org/wiki/Php>
5. MySQL [Электронный ресурс]. Дата обращения:5.06.2013. <http://ru.wikipedia.org/wiki/Mysql>
6. Plugin [Электронный ресурс]. Дата обращения:5.06.2013. <http://www.wisdomweb.ru/JQ/plugin.php>
7. AJAX [Электронный ресурс]. Дата обращения:5.06.2013. <http://ru.wikipedia.org/wiki/Ajax>
8. Сергей Трофимов. Использование моделей UML в процессе разработки программных систем [Электронный ресурс] . © Trofimov Sergey. Дата обращения: 8.06.2013. <http://www.caseclub.ru/articles/modeli.html>.
9. UML – Википедия [Электронный ресурс] . ©Wikimedia Foundation, Inc. Дата обращения: 8.06.2013. <http://ru.wikipedia.org/wiki/UML>
10. Главная страница [Электронный ресурс] . Дата обращения: 9.06.2013. <http://www.webeffector.ru/wiki/Главная_страница>
11. Организация работы за компьютером. [Электронный ресурс].Дата обращения: 10.06.2013.

<http://tvoykomputer.ru/organization-job-of-pc/>

# ПРИЛОЖЕНИЕ А – ЛИСТИНГ ПРОГРАММЫ

## Recipe.php

<?php

include 'header.php';

$id = (int)$\_GET['id'];

global $mysql;

$result1 = $mysql->query("select res.id, res.title, res.description,res.photo\_id, res.fulldescription,res.up, res.down, pic.`name` as photo, kit.`name` from recipe as res

inner join kitchen as kit

on res.kitchen\_id = kit.id

inner join picture as pic

on res.photo\_id = pic.id

where res.id = $id");

$result2 = $mysql->query("select ing.`name` as ingr, con.`count`, sys.`name` as sys from consist as con

inner join ingred as ing

on ing.id = con.id\_ingred

inner join `system` as sys

on sys.id = con.id\_system

where con.id\_recipe = $id");

$res = $mysql->query("select r.id, r.postedBy, r.putdate as putdate, u.username as username from recipe as r

inner join users as u

on r.postedBy = u.id

where r.id = $id");

function getDelete(){

global $mysql;

$result3 = $mysql->query("update `recipe` set exist=0 WHERE `id` = ".$\_REQUEST['id']);

if(!$result3) {

die("Error while deleting recipe from table! ".$mysql->error);

}

// $result4 = $mysql->query("delete from consist where id\_recipe = ".$\_REQUEST['id']);

// if(!$result4) {

// die("Error while deleting consist from table! ".$mysql->error);

// }

header('Location: category.php');

}

if (!empty($\_REQUEST)){

if (isset($\_POST['delete'])) {

getDelete();

}

}

$result = $mysql->query("SELECT \* FROM comment AS c

WHERE c.recipe\_id = {$id}

ORDER BY id LIMIT 5;");

if($isLoggedIn){

$resLike = $mysql->query("select recipe\_id from `like` where id\_user = {$\_SESSION['user\_id']} and recipe\_id = {$id}");

$count= count($resLike->fetch\_assoc());

}

?>

<div id="page">

<div id="page-bgtop">

<div id="page-bgbtm">

<div id="content">

<div class="post">

<h2 class="title"><a href="#">Welcome to Cooktasty </a></h2>

<div style="clear: both;">&nbsp;</div>

<div class="entry">

<p class="meta">

<?php $result3 = $res->fetch\_assoc(); ?>

<span class="date"><?php echo $result3['putdate']; ?></span>

<span class="posted">Автор рецепта: <a href="#"><?php echo $result3['username']; ?></a> </span>

</p>

<?php if($isLoggedIn && !($count)){ ?>

<div class="favorite" id="favorite">

<img src="css/images/fav.png" class="fav" id="fav">

<p class="favor">Добавить в избранное</p>

</div>

<?php } ?>

<?php $row1 = $result1->fetch\_assoc();?>

<h3><?php echo $row1['title']; ?> </h3>

<p class="border"><?php echo $row1['description']; ?></p>

<label for="kitch">Кухня: </label><p id="kitch" class="border"><?php echo $row1['name']; ?></p>

<img class="pic" src=" <?php echo $row1['photo']; ?> ">

<h4>Состав : </h4>

<div class="consistWhole">

<?php

while($row2 = $result2->fetch\_assoc()): ?>

<p class="consistPart"><?php echo $row2['ingr'] . ' ';

echo $row2['count'] . ' ';

echo $row2['sys'] . ' '; ?></p>

<?php endwhile; ?>

</div>

<h4>Приготовление : </h4>

<p class="border full"><?php echo $row1['fulldescription']; ?></p>

</div>

<div class="edit">

<?php

if (isset ($\_SESSION['current\_user'])){

if ( ($currentUser['is\_admin'] == 1) || ($\_SESSION['user\_id'] == $result3['postedBy']) ) { ?>

<p><a href="edit.php?id=<?=$\_GET['id']?>">Редактировать рецепт!</a></p>

<form method="post">

<div>

<button type="submit" value="Удалить" name="delete">Удалить</button>

</div>

</form>

<?php

}

}

?>

</div>

<div class="box1">

<?php $up = $row1['up'];

$down = $row1['down'];?>

<div class='up'>

<a href="" class="vote" name="up"><?php echo $up; ?></a>

</div>

<div class='down'>

<a href="" class="vote" id="<?php echo $id; ?>" name="down"><?php echo $down; ?></a>

</div>

</div>

<div class="commentaries">

<div id="comment\_list">

<?php while ($comment = $result->fetch\_array())

{

$text = $comment["text"];

$author = $comment["author"];

$date = $comment["date"];

?>

<div class="com">

<h4 class="aut"><?php echo $author . " " . $date;?></h4>

<p class="txt"><?php echo $text; ?></p>

</div>

<?php } ?>

</div>

<div id="load">

<img src="css/images/loading.gif" id="imgLoad">

<div><p>Загрузить еще</p></div>

</div>

<div class="add">

<div id="cerror"></div>

<form action="#" method="POST" enctype="multipart/form-data" name="addcom" id="addcom" onSubmit="return false">

<p>Ваше имя:</p>

<input name="author" type="text" class="pole inp" id="author">

<p>Текст комментария:</p>

<textarea name="text" class="inp"></textarea>

<input name="id" type="hidden" value="<?php echo $id; ?>">

<input class="adscom bt" id="add\_comment" name="button" type="button" value="Добавить комментарий">

</form>

</div>

</div>

</div>

<div style="clear: both;">&nbsp;</div>

</div>

<!-- end #content -->

<?php include 'sideBar.php';?>

<!-- end #sidebar -->

</div>

<p id="back-top">

<a href="#page-bgtop"><span></span></a>

</p>

</div>

<div style="clear: both;">&nbsp;</div>

</div>

<!-- end #page -->

</div>

<?php include 'footer.php';?>

<script type="text/javascript">

$('#add\_comment').click(

function(){

$.ajax({

type:'POST',

dataType:'json',

url: 'comment\_new.php',

data: $('#addcom').serialize(),

success: function(data) {

var object = eval(data);

$('#comment\_list').append(object.block);

console.log(object);

},

error: function(){

console.log('error');

}

});

});

$(document).ready(function(){

$("#imgLoad").hide(); // Скрываем прелоадер

});

var num = 5; //чтобы знать с какой записи вытаскивать данные

$(function() {

$("#load div").click(function(){ // Выполняем если по кнопке кликнули

$("#imgLoad").show(); // Показываем прелоадер

$.ajax({

url: "action.php", // Обработчик

type: "GET", // Отправляем методом GET

data: {"num": num, id: <?php echo $\_GET['id'] ?> },

cache: false,

success: function(response){

if(response == 0){ // Смотрим ответ от сервера и выполняем соответствующее действие

alert("Больше нет записей");

$("#imgLoad").hide();

}else{

$("#comment\_list").append(response);

num = num + 5;

$("#imgLoad").hide();

}

}

});

});

});

$(function() {

$(".vote").click(function()

{

<?php if($isLoggedIn){ ?>

//var id = $(this).attr("id");

var name = $(this).attr("name");

//var dataString = 'id='+ id ;

var parent = $(this);

if(name=='up')

{

$.ajax({

type: "GET",

url: "up\_vote.php",

data: { user\_id : <?php echo $\_SESSION['user\_id'] ?>, id : <?php echo $\_GET['id'] ?> },

success: function(html)

{

parent.html(html);

} });

}

else

{

$.ajax({

type: "GET",

url: "down\_vote.php",

data: { user\_id : <?php echo $\_SESSION['user\_id'] ?>, id : <?php echo $\_GET['id'] ?> },

success: function(html)

{

parent.html(html);

}

});

}

<?php }

else { ?>

alert("Зарегистрируйтесь сначала!");

<?php } ?>

return false;

});

});

$(document).ready(function(){

// hide #back-top first

$("#back-top").hide();

// fade in #back-top

$(function () {

$(window).scroll(function () {

if ($(this).scrollTop() > 350) {

$('#back-top').fadeIn();

} else {

$('#back-top').fadeOut();

}

});

// scroll body to 0px on click

$('#back-top a').click(function () {

$('body,html').animate({

scrollTop: 0

}, 800);

return false;

});

});

});

<?php if($isLoggedIn){ ?>

$(document).ready(function(){

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

$.ajax({

type: "GET",

url: "favorite.php",

data : { user\_id : <?php echo $\_SESSION['user\_id'] ?>, id : <?php echo $\_GET['id'] ?> },

success: function(response){

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

alert("Added to favorite");

},

error: function(){

console.log('error');

}

});

});

});

<?php } ?>

</script>