

КУРСОВАЯ РАБОТА

на тему: **Веб-сайт ресторана на платформе ASP.NET**

Студент

Руководитель

2023

Оглавление

Введение.....	
Глава 1 Анализ бизнес-процессов ООО «Честер»	
1.1 Краткая характеристика подразделения и его видов деятельности и сущность задачи автоматизации.....	
1.2. Обоснование выбора методологии и технологии проектирования Web- представительства.....	
1.3 Концептуальное проектирование Web-представительства ресторана....	
1.3.1 Разработка и анализ модели «КАК ЕСТЬ» бизнес-процесса управления заказами ООО «Честер».....	
1.3.2 Анализ недостатков существующего бизнес-процесса и рекомендации по его усовершенствованию с помощью интернет-технологий	
1.3.3 Разработка модели «КАК ДОЛЖНО БЫТЬ» бизнес- процесса управления заказами ООО «Честер».....	
1.5 Анализ известных решений Web-представительств.....	
Глава 2 Разработка web-представительства ООО «Честер»	
2.1 Проектирование базы данных Web-представительства.....	
2.1.1 Диаграмма вариантов использования бизнес-процесса управления заказами клиентов.....	
2.1.2 Диаграмма классов Web-представительства.....	
2.1.3 Диаграмма последовательности бизнес-процесса управления заказами	
2.2 Разработка логической модели данных Web-представительства.....	
2.3 Выбор архитектуры Web-представительства	
2.4 Выбор средств реализации Web-представительства.....	
2.4.1 Выбор системы управления базами данных.....	
2.4.2 Физическая модель данных Web-представительства.....	
2.4.3 Выбор языка программирования Web-приложения.....	

2.5	Описание программных модулей.....
2.6	Обоснование экономической эффективности разработки Web-представительства ООО «Честер»
	Заключение
	Список используемой литературы
	Приложение.....

Введение

В сфере общественного питания все большую роль приобретают уровень и качество автоматизации основных бизнес-процессов, что создает предпосылки для экономии средств, времени и гибкости управления и конкурентные преимущества для выживания предприятия в условиях современного экономического кризиса.

Проблема некачественного обслуживания клиентов особенно чувствительна для ставших популярными в последнее время ресторанов быстрого питания.

Заполнение бланков заказов отнимает дополнительное время, которое работники ресторана могли использовать более эффективно. Наличие большого количества бумажных источников информации ведет к частым ошибкам и неточностям в расчетах, что вызывает недовольство клиентов и негативно влияет на имидж ресторана.

В ООО «Честер» пришли к выводу о необходимости решения данной проблемы путем разработки и внедрения информационной онлайн-системы управления заказами клиентов ресторана на основе собственного Web-представительства, которое помимо выполнения маркетинговых функций будет обеспечивать менеджмент ресторана информацией, необходимой для принятия управленческих решений.

Глава 1 Анализ бизнес-процессов ООО «Честер»

1.1 Идентификация ресторана быстрого питания

ООО «Честер» (далее – ресторан) является рестораном быстрого питания г. Москва.

Адрес ресторана: г. Тольятти, ул. 40 лет Победы, 69.

Компания предоставляет услуги общественного питания через принадлежащие ей предприятия быстрого обслуживания.

Форма собственности: частная, собственность акционеров.
Организационно – правовая форма: транснациональная корпорация (акционерное общество открытого типа).

Форма управления – иерархическая структура.

Основные принципы управления:

- четкое разделение труда способствует появлению высококвалифицированных специалистов в каждом деле;
- строгая иерархия уровней управления, при которой действия нижестоящего звена управления контролируются вышестоящим;
- формальная обезличенность руководителей, обязанных следовать инструкциям и предписаниям по занимаемой должности;
- найм на работу в строгом соответствии с квалификацией работника и его увольнение «строго по закону».

В таблице 1.1 приведена структура должностей ООО «Честер»

Таблица 1.1 - Структура должностей ресторана ООО «Честер»

Непосредственно в процессе обслуживания заказов участвуют члены бригады ресторана.

В настоящее время ООО Честер не имеет собственного Web представительства.

Уровень	Должность
6	Директор
5	1-й ассистент директора
4	2-й ассистент директора
3+	менеджер-стажер
3	менеджер производственного участка
2	инструктор по обучению
1	работник ресторана – член бригады

Обслуживание ИТ-инфраструктуры ООО «Честер» осуществляется по модели аутсорсинга: в ресторане не предусмотрена собственная ИТ-служба, и при возникновении проблем с вычислительной и оргтехникой руководство ресторана обращается за помощью в ИТ-компанию.

1.1.1 Краткая характеристика подразделения и его видов деятельности и сущность задачи автоматизации

Работа выполнена для отдела операционного менеджмента ресторана.

В задачи отдела входит управление производственным процессом в ресторане.

Основной штатной единицей подразделения является менеджер производственного участка, непосредственно отвечающий за Высокий уровень обслуживания клиентов – главная стратегическая задача ООО «Честер».

Одной из проблем ресторана является снижение производительности выполнения заказов клиента, обусловленное, в том числе отсутствием соответствующего программного обеспечения.

Задача автоматизации состоит во внедрении Web-представительства, обеспечивающего помимо решения маркетинговых задач информационную поддержку бизнес-процесса управления заказами клиентов ресторана [9].

1.2 Обоснование выбора методологии и технологии

В настоящее время наиболее распространены методологии проектирования, опирающиеся на каскадную, итерационную и спиральную модель жизненного цикла (ЖЦ) информационной системы (ИС) [1].

Сравним достоинства и недостатки каждой модели (таблица 1.2). Таблица 1.2 - Достоинства и недостатки моделей ЖЦ ИС

Модель ЖЦ ИС	Достоинства	Недостатки
Каскадная	Выполняемые в четкой последовательности стадии проектирования позволяют уверенно планировать сроки выполнения работ и соответствующие ресурсы	Результаты разработки доступны заказчику только в конце проекта
Итерационная	Позволяет переходить на следующий этап, не дожидаясь полного завершения работы на текущем этапе	Сложность определения момента перехода на следующий этап
Спиральная	Позволяет быстрее показать пользователям системы работоспособный продукт, тем самым, активизируя процесс уточнения и дополнения требований	Затруднены операции временного и ресурсного планирования всего проекта в целом

На основании результатов сравнения выбрана для разработки Web-представительства ресторана методология проектирования, опирающаяся на спиральную модель ЖЦ ИС и модельно-ориентированное проектирование, так как представляется наиболее оптимальной для выполнения краткосрочных проектов.

В данной методологии на каждом витке спирали выполняется создание очередной версии ИС, уточняются требования проекта, определяется его качество, планируются работы следующего витка. Особое внимание уделяется начальным этапам разработки – анализу и проектированию, где реализуемость тех или иных технических решений проверяется и обосновывается посредством создания прототипов ИС.

Модельно-ориентированное проектирование заключается в адаптации состава и характеристик типовой ИС в соответствии с трехуровневым представлением модели объекта автоматизации: концептуальным, логическим и физическим уровнями.

1.3 Концептуальное проектирование Web- ресторана

На данном этапе используются методологии, основанные на структурном подходе и концепции реинжиниринга [10].

Первая стадия анализа – структурный анализ предприятия – начинается с исследования того, как организована система управления рестораном, с обследования функциональной и информационной структур системы управления, определения существующих и возможных потребителей информации.

По результатам обследования аналитик на первой стадии анализа выстраивает обобщенную модель исходной предметной области, отображающую ее функциональную структуру, особенности основной деятельности и информационное пространство, в котором эта деятельность осуществляется.

На этом материале аналитик строит функциональную модель ресторана «КАК ЕСТЬ» (AS-IS).

Вторая стадия работы, к которой обязательно привлекаются заинтересованные представители заказчика, а при необходимости и независимые эксперты, состоит в анализе модели «КАК ЕСТЬ», выявлении ее недостатков и узких мест, определении путей совершенствования системы управления на основе выделенных критериев качества.

Третья стадия анализа – создание усовершенствованной обобщенной логической модели, отображающей реорганизованную предметную область или ее часть, которая подлежит автоматизации – модель «КАК ДОЛЖНО БЫТЬ» (TO-BE).

Заканчивается процесс разработкой концептуальной модели реорганизованной предметной области, на которой обязательно обозначены границы автоматизации ресторана.

В структурном анализе используются в основном две группы средств, иллюстрирующих функции, выполняемые системой и отношения между данными [11].

Каждой группе средств соответствуют определенные виды моделей (диаграмм), наиболее распространенными, среди которых являются следующие:

- IDEF0–модели и соответствующие функциональные диаграммы;
- DFD–диаграммы потоков данных.

Перечисленные модели в совокупности дают полное функциональное описание ИС независимо от того, является ли она существующей или вновь разрабатываемой.

Состав диаграмм в каждом конкретном случае зависит от необходимой полноты описания системы.

Для повышения эффективности процесса структурного моделирования системы рекомендуется использовать доступные CASE – средства, например, триал-версию программного продукта BPWin или бесплатно распространяемый программный продукт Ramus.

1.3.1 Разработка и анализ модели «КАК ЕСТЬ» бизнес-процесса управления заказами ООО «Честер»

Компания «Честер» организовала региональную сеть ресторанов по собственной референтной модели. Бизнес-процесс управления заказами организован следующим образом: – Клиент обращается к Работнику ресторана с Заказом;

– Работник ресторана на Терминале формирует Бланк заказа и счет на его оплату;

– после оплаты Клиентом счета Работник ресторана выдает Клиенту чек и Номер заказа;

– после получения заказа Клиентом Работник ресторана делает отметку о его выполнении на Терминале.

Оформление заказа регламентируются Законом РФ «О защите прав потребителей».

Управление заказами регламентируется Инструкцией по управлению заказами клиентов ресторана.

Модель бизнес-процесса «КАК ЕСТЬ» описывает существующие принципы организации бизнес-процесса управления маркетингом и управления заказами ООО «Честер» . На рисунках 1.1-1.3 представлена модель «КАК ЕСТЬ» бизнес-процесса управления заказами ООО «Честер» с точки зрения Менеджера производственного участка компании.

Для разработки диаграмм модели использованы методологии IDEF0 и DFD.



Рисунок 1.1 – Контекстная диаграмма бизнес-процесса управления заказами ООО «Честер» (г.Тольятти) «КАК ЕСТЬ» в методологии IDEF0 (0-й уровень)

На представленной диаграмме изображены следующие элементы:

- входные данные: Заказ клиента;
- выходные данные: Чек, Отметка о выполнении заказа;
- управляющие воздействия: Закон РФ «О защите прав потребителей», Инструкция по управлению заказами клиентов;
- исполнители: Клиент, Работник ресторана, Терминал.

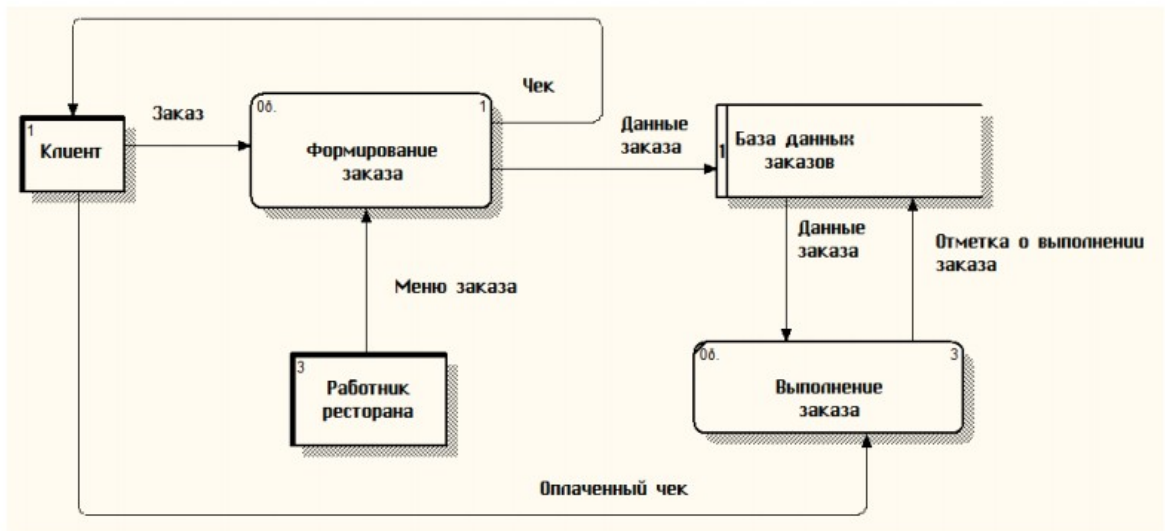


Рисунок 1.2 – DFD-декомпозиция бизнес-процесса управления заказами ООО «Честер» (г.Тольятти) «КАК ЕСТЬ» (1-й уровень)



Рисунок 1.3 – DFD-декомпозиция процесса «Формирование заказа» «КАК ЕСТЬ» (2-й уровень)

Данная модель является основой для анализа и дальнейшего совершенствования бизнес-процесса управления заказами клиентов ресторана.

1.3.2 Анализ недостатков существующего бизнес-процесса и рекомендации по его усовершенствованию с помощью интернет-технологий

Анализ модели «КАК ЕСТЬ» показал, что существующий бизнес-процесс малоэффективен и имеет следующие недостатки:

- невысокая эффективность маркетинга ресторана, обусловленная отсутствием собственного Web-представительства;
- отсутствие возможности онлайн-режима ввода заказов клиентами.

С учетом вышеизложенного принято решение улучшить бизнес-процесс путем разработки и внедрения Web-представительства ООО «Честер», обеспечивающего возможность онлайн-режима ввода заказов клиентами.

1.3.3 Разработка модели «КАК ДОЛЖНО БЫТЬ» бизнес-процесса управления заказами ООО «Честер»

На рисунках 1.4-1.6 представлена модель «КАК ДОЛЖНО БЫТЬ» бизнес-процесса управления заказами ООО «Честер» (разработанная с помощью методологий IDEF0 и DFD).

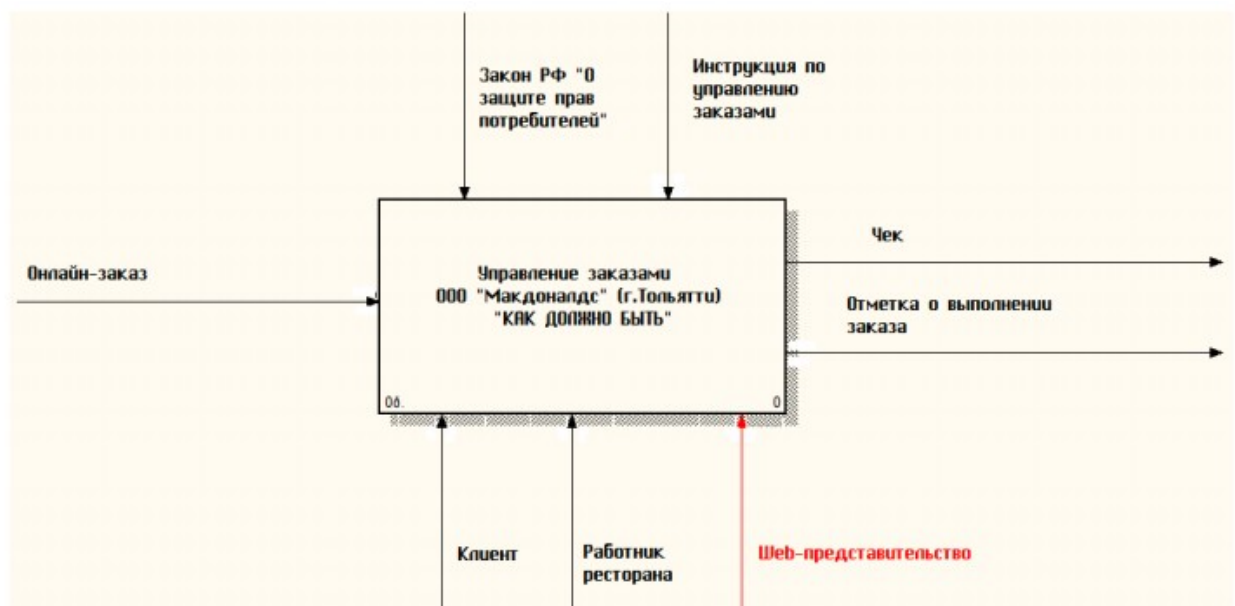


Рисунок 1.4 – Контекстная диаграмма бизнес-процессов управления заказами ООО «Честер» «КАК ДОЛЖНО БЫТЬ» (0-й уровень)

На представленной диаграмме изображены следующие элементы:

- входные данные: Онлайн-заказ;
- выходные данные: Чек, Отметка о выполнении заказа;
- управляющие воздействия: Закон РФ «О защите прав потребителей», Инструкция по управлению заказами клиентов;
- исполнители: Клиент, Работник ресторана, Web-представительство.

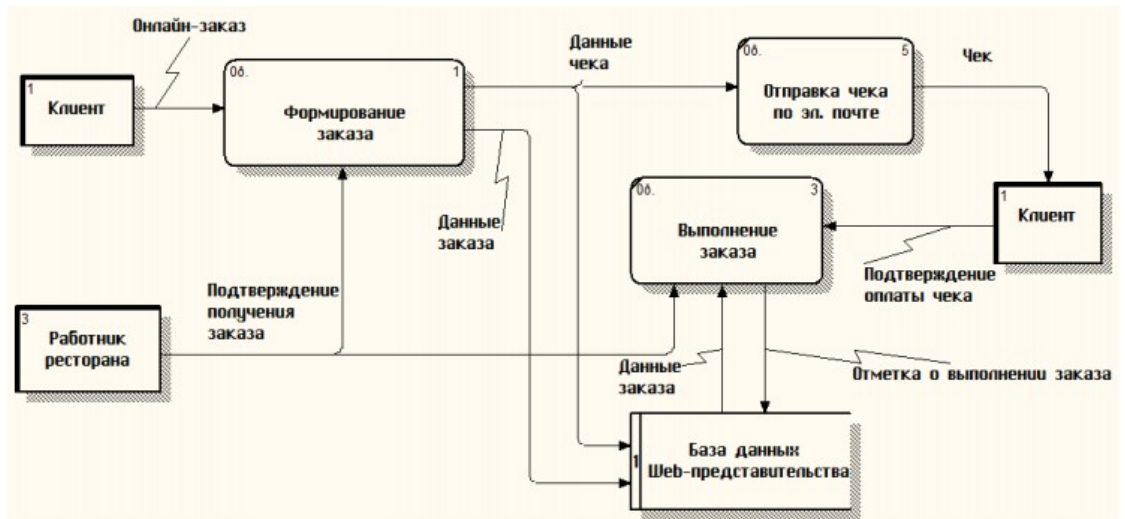


Рисунок 1.5 – DFD-диаграмма бизнес-процесса управления заказами ООО «Честер» «КАК ДОЛЖНО БЫТЬ» (1-й уровень)

Новые и измененные элементы выделены красным цветом.

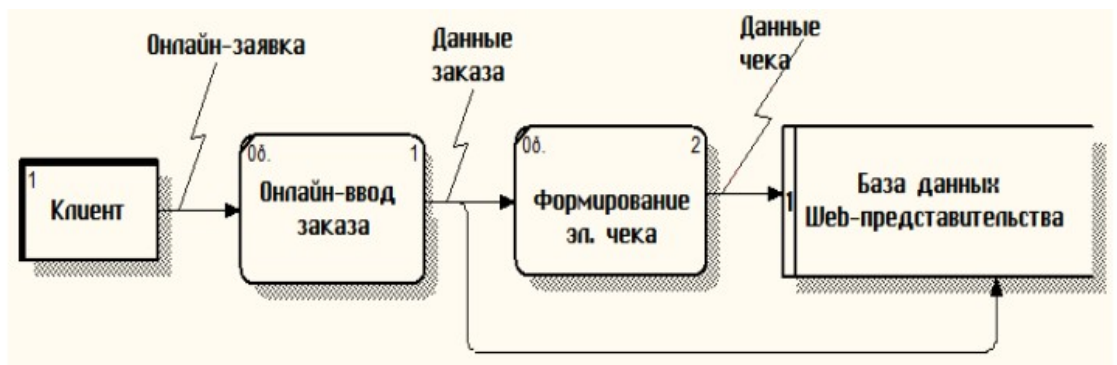


Рисунок 1.6 – DFD-декомпозиция процесса «Формирование заказа» «КАК ДОЛЖНО БЫТЬ» (2-й уровень)

Таким образом, усовершенствование исследуемого бизнес-процесса достигается путем разработки и внедрения Web-представительства, отвечающего требованиям заказчика.

Глава 2 Разработка web-представительства ООО

«Честер».

2.1 Проектирование базы данных Web-представительства

Логическое моделирование предназначено для создания объектной модели программной архитектуры системы и логической модели данных.

Для обоснования и постановки задачи на разработку логической модели данных Web-представительства используется методология объектно-ориентированного проектирования и анализа, основанная на языке UML [12].

В концепции UML для описания системы с различных точек зрения используются три типа моделей: функциональная, модели классов и модель взаимодействий.

Полное описание системы требует наличия всех трех моделей.

Каждая модель применяется на всех этапах проектирования и строится с помощью специальных диаграмм языка UML.

2.1.1 Диаграмма вариантов использования бизнес-процесса управления заказами клиентов

На первом этапе строим диаграмму вариантов использования, которая представлена на рисунке 2.1.

Диаграммы вариантов использования (use case diagram) применяются при бизнес-анализе для моделирования видов работ, выполняемых организацией, и для моделирования функциональных требований к проектируемой системе при ее проектировании и разработке. Диаграмма вариантов использования строится на основе DFD-декомпозиции бизнес-процесса управления заказами «КАК ДОЛЖНО БЫТЬ», представленной на рисунке 1.5.

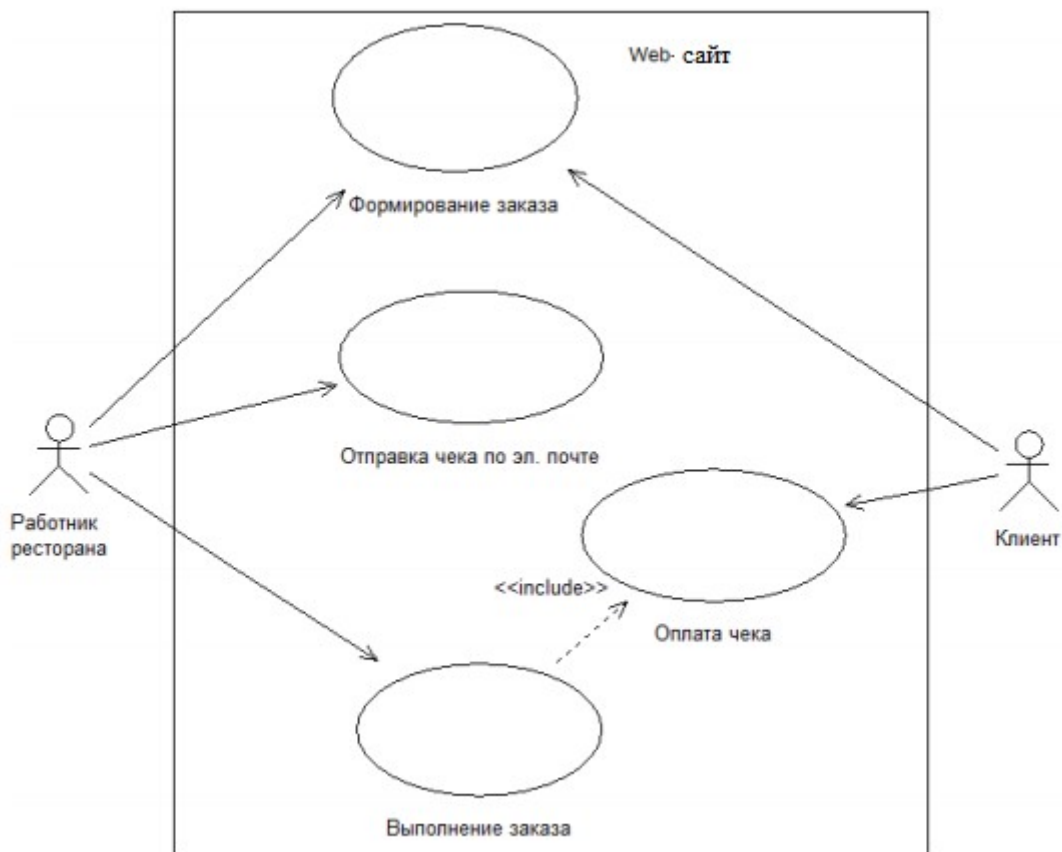


Рисунок 2.1 – Диаграмма вариантов использования бизнес-процесса управления заказами клиентов ООО «Честер» «КАК ДОЛЖНО БЫТЬ»

Диаграмма вариантов использования описывает функциональный аспект рассматриваемой информационной системы «КАК ДОЛЖНО БЫТЬ», предоставляя дополнительную информацию об отношениях между различными вариантами использования и внешними пользователями-актерами.

2.1.2 Диаграмма классов Web-представительства

Диаграмма классов описывает статическую структуру объектов системы и их отношения. Эта модель определяет контекст разработки программы, то есть предметную область. Модель классов изображается на диаграммах классов. Диаграмма классов – это граф, вершинами которого являются классы, а ребрами – их отношения.

Построение диаграммы классов системы производится следующим образом:

- в описании классов выделяются кандидаты в классы – существительные, которые потенциально могут соответствовать классам (при этом следует помнить, что существительные могут относиться к объектам, ассоциациям или атрибутам классов);

- анализируются роли кандидатов в системе. Каждый класс должен выполнять некоторые действия и взаимодействовать с другими классами. Он должен иметь также уникальное имя, отражающее характер абстракции, представляемой данным классом.

На рисунке 2.2 изображена диаграмма классов Web-представительства.

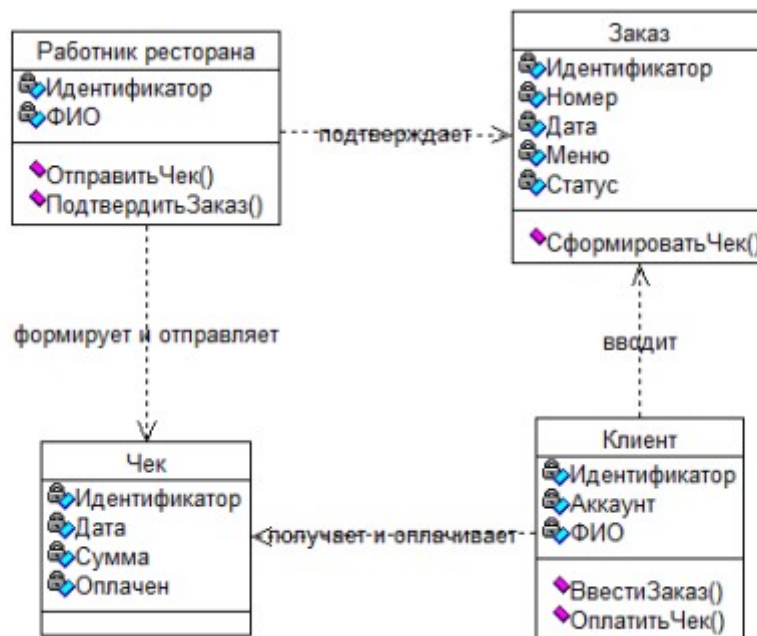


Рисунок 2.2 – Диаграмма классов Web-представительства

Спецификация классов:

- Клиент – класс объектов-пользователей Web-представительства, выполняющих ввод онлайн-заказов;

- Работник ресторана – класс объектов-пользователей Web-представительства, формирующих чеки и обслуживающих заказы клиентов;

- Заказ – класс объектов-документов;

- Чек – класс объектов-документов.

Представленная диаграмма предназначена для разработки объектной

модели приложения Web-представительства.

2.1.3 Диаграмма последовательности бизнес-процесса управления заказами

Диаграмма последовательности (sequence diagram) отображает динамический аспект системы, представлена на рисунке 2.3.

В рамках динамического аспекта реализуется ситуационный подход, позволяющий решить следующие задачи:

- определение способов выполнения процесса в зависимости от конкретной ситуации. Выбор способа реализации (сценария) определяется с привлечением того или иного ответственного исполнителя (актера);
- определение взаимодействия организационного, функционального и элементного аспектов;
- определяется документооборот (обмен сообщениями) между взаимодействующими исполнителями.



Рисунок 2.3 – Диаграмма последовательности бизнес-процесса управления заказами клиентов ресторана

В случайный момент времени объект *Клиент* отправляет объекту *Работник ресторана* сообщение «Открыть заказ».

Объект *Работник ресторана* формирует чек и отправляет его объекту *Клиент* по электронной почте с сообщением «Оплатить чек».

Объект *Клиент* оплачивает чек и отправляет объекту *Работник ресторана* подтверждение оплаты чека.

Объект *Работник ресторана* сообщает объекту *Клиент* о подтверждении заказа.

2.2 Разработка логической модели данных Web-сайта

Логическая модель данных является начальным прототипом будущей базы данных.

Логическая модель данных – описание объектов предметной области, их атрибутов и взаимосвязей между ними в том объеме, в котором они подлежат непосредственному хранению в базе данных системы [8].

Логическая модель данных строится в терминах информационных единиц, но без привязки к конкретной СУБД.

Логическая модель данных строится на основе диаграммы классов, представленной на рисунке 2.2, которая трансформируется в ER-модель («сущность-связь»).

Затем в модель добавляются новые сущности, необходимые для обеспечения необходимого уровня нормализации данных.

На рисунке 2.4 изображена логическая модель данных Web-представительства, разработанная в методологии IDEF1X.

Как следует из рисунка, в процессе разработки выделены следующие основные сущности:

- Клиент;
- Заказ;
- Работник ресторана;
- Чек

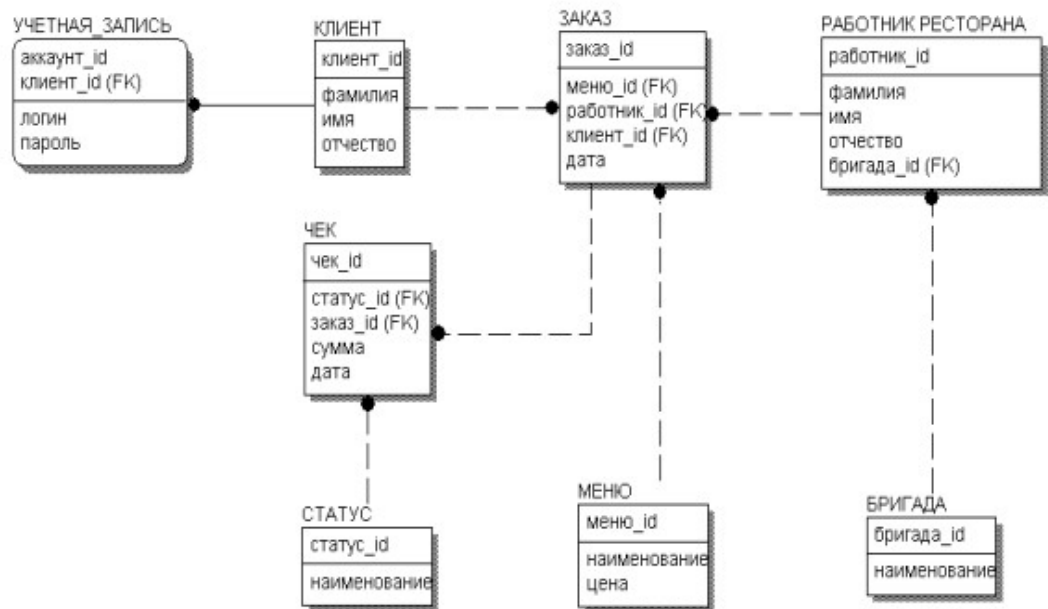


Рисунок 2.4 – Логическая модель данных Web-представительства

Дополнительные сущности Учетная запись, Статус, Меню и Бригада введены для приведения модели к уровню нормальной формы Бойса-Кодда.

Между основными сущностями логической модели установлены следующие связи:

- Клиент может сделать несколько Заказов («один ко многим»);
- Работник ресторана может обслужить несколько Заказов («один ко многим»);
- По каждому заказу может быть сформирован только один Чек («один к одному»).

Все связи между основными сущностями неидентифицирующие.

Представленная логическая модель данных является концептуальной моделью данных Web-представительства и будет использована в качестве основы при проектировании реляционной модели базы данных системы на стадии физического моделирования данных.

2. 3. Выбор архитектуры Web-представительства

Информационные системы, разрабатываемые с помощью Web-технологий, реализуются в архитектуре «клиент-сервер» [4].

Архитектура «клиент-сервер» может быть представлена в следующих вариантах:

1. Традиционная двухзвенная архитектура «клиент - сервер».

Первое звено – клиентские компьютеры с прикладными программами, с помощью которых пользователи обращаются по сети к базе данных.

Второе звено – сервер баз данных, также участвующий в обработке данных.

Благодаря двухзвенной архитектуре снижается нагрузка на информационную сеть, поскольку передаются только запросы и ответы на них.

Преимущества:

- распределение функций вычислительной системы между несколькими независимыми компьютерами в сети, это позволяет упростить обслуживание вычислительной системы, в частности, замена, ремонт, модернизация или перемещение сервера, не затрагивают клиентов;

- все данные хранятся на сервере, который, как правило, защищён гораздо лучше большинства клиентов;

- на сервере проще обеспечить контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа.

Недостатки:

- ограниченность применения для Web-приложений;

- высокая стоимость оборудования клиентов.

2. Трёхзвенная архитектура «клиент - сервер».

Первое звено – клиентская программа. Клиент посылает запросы на выполнение нужных действий. Вся логика системы реализуется на других уровнях, поэтому требования к клиентской машине минимальные. Такие клиентские программы называют тонкими клиентами. В качестве базового программного обеспечения используются Web-браузеры- IE, Mozilla и другие.

Второе звено – сервер приложений – программное обеспечение промежуточного уровня. Синхронизирует работу компонентов всей системы и организует связь между ними. Обычно такую роль играют сетевые службы и

приложения, в том числе Web-серверы.

Третье звено – система управления базами данных (СУБД), относящаяся к категории серверов баз данных. Данный сервер не работает напрямую с клиентскими программами, что повышает безопасность информации в системе.

Такая архитектура разбивает процесс обработки данных между клиентом, сервером приложений и хранилищем данных. В отличие от традиционной двухзвенной архитектуры здесь присутствует сервер приложений, как промежуточное звено между клиентом и хранилищем данных.

В системе присутствует два хранилища. В первом (обычно реляционная база данных) хранятся все данные, которые публикуются на сайте. Во втором (обычно файловая система) хранятся элементы представления – шаблоны, графические изображения и т.д.

Получая запрос, сервер приложений обрабатывает его, связываясь с хранилищем данных, в каком бы месте необходимые данные не находились.

Клиент лишь получает результат в виде HTML-файла. Таким образом, сервер приложений является стандартизированной платформой для динамической доставки контента и построения основных приложений.

Серверов приложений может быть много, а связь с ними происходит через Web-сервер.

В таблице 2.1 приведены результаты сравнительного анализа архитектур Web-представительства.

Архитектура/ Требование (баллы)	Клиент-сервер, двухзвенная модель	Клиент- сервер трехзвенная модель
масштабируемость	-	+
оптимизация сетевого трафика	+	+
простота реализации	+	-
поддержка современных Web-технологий	-	+
Сумма баллов	2	3

По результатам анализа при реализации Web-представительства выбрана трехзвенная архитектура «клиент-сервер».

2.4. Выбор средств реализации Web-представительства

2.4.1 Выбор системы управления базами данных

При выборе СУБД учитывалось требование заказчика обеспечить низкую стоимость владения Web-представительства.

Рассмотрим бесплатно распространяемые СУБД MySQL, FireBird, Oracle Database XE и произведем их сравнительный анализ.

MySQL – это реляционная СУБД. MySQL характеризуется большой скоростью, устойчивостью и легкостью в использовании, является решением для малых и средних приложений. Это одна из самых быстрых современных СУБД [20].

Firebird (FirebirdSQL) – компактная, кроссплатформенная, свободная система управления базами данных (СУБД), работающая на Linux, Microsoft Windows и разнообразных Unix платформах.

В качестве преимуществ Firebird можно отметить многоверсионную архитектуру, обеспечивающую параллельную обработку оперативных и аналитических запросов (это возможно, потому что читающие пользователи не блокируют пишущих), компактность (дистрибутив 5Mb), высокую эффективность и мощную языковую поддержку для хранимых процедур и триггеров.

Oracle Database 10g Express Edition (Oracle Database XE) – бесплатная версия СУБД для разработчиков.

Oracle Database XE предоставляет те же интерфейсы SQL и PL/SQL, что и во всех остальных версиях Oracle Database 10g, а также широкий спектр программных интерфейсов. Предоставляется полная поддержка разработки и развертывания приложений для разработчиков, работающих на платформах Java, .NET, PHP и Windows.

Максимальный размер БД – 5 Гб.

Сравним эти СУБД по основным критериям оценки серверов баз данных, результаты сравнения представлены в таблице 2.2.

Таблица 2.2 - Сравнительный анализ СУБД

Характеристики	MySQL	FireBird	Oracle Database XE
скорость работы	+	-	+
масштабируемость	+	+	-
простота администрирования	+	+	-
Сумма баллов	3	2	1

Из сравнительной таблицы можно сделать вывод, что СУБД MySQL, обеспечивает лучшие характеристики, поэтому выбрана именно она в качестве сервера баз данных Web-представительства.

Самой популярной версией данной СУБД в настоящее время является MySQL 5.x.

2.2.2 Физическая модель данных Web-представительства

Физическая модель данных – логическая модель данных, выраженная в терминах языка описания данных конкретной СУБД.

Физическая модель данных содержит все детали, необходимые конкретной СУБД для создания базы: наименования таблиц и столбцов, типы данных, определения первичных и внешних ключей и т.п.

Физическая модель данных строится на основе логической модели данных с учетом ограничений, накладываемых особенностями архитектуры и типизации данных СУБД MySQL.

На рисунке 3.1 представлена физическая модель данных Web-представительства, построенная на основе вышеперечисленных рекомендаций в методологии IDEF1X.

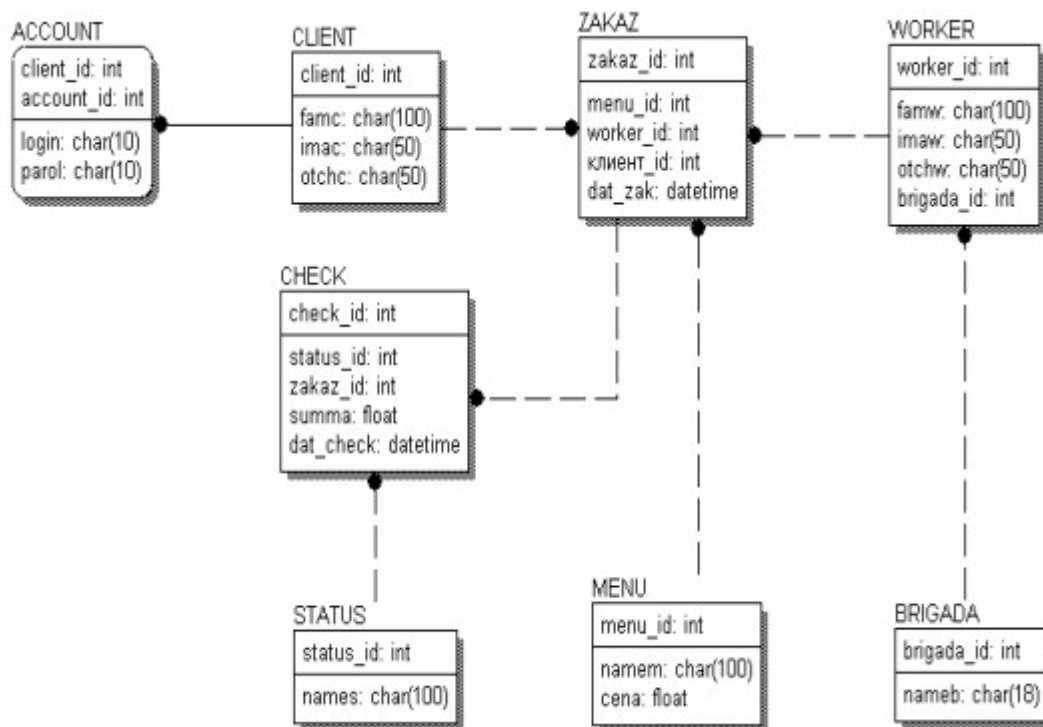


Рисунок 2.5 – Физическая модель данных Web-сайта

В базе данных, реализованной на основе представленной модели, используется нормальная форма Бойса-Кодда.

2.2.3 Выбор языка программирования Web-приложения

В настоящее время для разработки динамических сайтов широко применяются такие технологии Web-программирования, как ASP.NET, Perl и PHP [19].

ASP.NET – технология создания веб-приложений и веб-сервисов от компании Майкрософт. Она является составной частью платформы Microsoft .NET и развитием более старой технологии Microsoft ASP. ASP.NET опирается на многоязыковые возможности .NET, что позволяет писать код страниц на VB.NET, Delphi.NET, Visual C#, J# и т. д.

Perl – высокоуровневый интерпретируемый динамический язык программирования общего назначения. Основной особенностью языка считаются его богатые возможности для работы с текстом, в том числе, работа с регулярными выражениями, встроенная в синтаксис. Perl унаследовал

много свойств от языков Си, AWK, скриптовых языков командных оболочек UNIX.

PHP – свободно распространяемый язык Web-программирования. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических Web-сайтов [15].

Результаты сравнительного анализа языков Web-программирования сведены в таблице 2.3.

Таблица 2.3 -Сравнительный анализ языков Web- программирования

Характеристики	ASP.NET	Perl	PHP
поддержка объектно-ориентированной парадигмы	+	+	+
свободно распространение среды разработки	+	+	+
простота интеграции с СУБД MySQL	+	-	-
знание разработчиком	+	-	+
Сумма баллов	4	2	1

По результатам анализа выбрана технология ASP.NET.

ASP.NET является объектно-ориентированной моделью разработки Web-приложений. Сами ASP.NET-страницы являются объектами классов. Можно создавать программный код с возможностью его повторного использования классами. Эти классы можно использовать для создания экземпляров объектов.

Объектная модель — это иерархия объектов, предоставляющих разработчикам определенные возможности. В ASP.NET используется новая структура Web-страниц, которая отличается от структуры ASP-страниц и обеспечивает поддержку объектной модели для сохранения содержимого ASP.NET-страницы. Добавлен новый класс элементов управления — серверные элементы управления. Можно добавлять собственные комментарии и связывать эти данные с серверными элементами управления. Для оформления Web-

страниц имеются наборы директив, которые предназначены для установки параметров. Например, параметры `TraceContext` и `isEnabled` позволяют, соответственно, включить и отключить механизм отслеживания Web-запросов.

ASP.NET определяет шесть внутренних объектов структуры страниц:

- `application`;
- `ObjectContext`;
- `response`;
- `request`;
- `server`;
- `session`.

Эти объекты встроены в модель ASP.NET-страниц и готовы к использованию.

В объекте `application` хранятся данные, которые будут доступны всем пользователям, одновременно работающим с Web-приложением.

Данные о сеансе работы каждого отдельного пользователя сохраняет объект `session`. Объект `application` используется для хранения глобальных переменных, которые содержат информацию, общедоступную для всех пользователей Web-приложения, например, приветственное сообщение или индекс посещения Web-страницы. Этот объект представляет собой коллекцию разнородных элементов. Пользователи совместно используют объекты Web-приложения, поэтому требуется гарантировать, что несколько пользователей не смогут одновременно изменять переменные, хранимые в объекте `application`. Для блокирования содержимого коллекции объектов от изменения применяется метод `Lock`, для разблокирования — метод `Unlock`. Так же существуют методы `Contents.Remove` и `Contents.RemoveAll`, которые удаляют один элемент из семейства `Contents` или все сразу соответственно.

С помощью объекта `ObjectContext` выполняется фиксация или откат транзакций, управляемых MTS. Транзакции могут быть инициированы со страницы ASP.NET. Методы `SetComplete` и `SetAbort` объекта `ObjectContext` используются, соответственно, для фиксации и отката транзакций.

Объект `response` можно использовать для передачи выходной информации клиенту. Методы объекта `response`:

- `AddHeader` – устанавливает заголовок HTML имя равным значению.
- `AppendToLog` – добавляет строку в конец записи журнала веб-сервера, относящейся к этому запросу.
- `BinaryWrite` – записывает предоставленную информацию в текущий вывод HTTP без преобразования наборов символов.
- `Clear` – стирает любой буферизованный вывод HTTP.
- `End` – останавливает обработку файла `.asp` и возвращает текущий результат.
- `Flush` – немедленно передает буферизованный вывод.
- `Redirect` – отправляет обозревателю сообщение о перенаправлении, вызывая попытку обозревателя подключиться к другому URL.
- `Write` – записывает переменную в виде строки в текущий вывод HTTP.

В объекте `request` сохраняется информация, отправляемая браузером клиента на сервер в HTTP-запросе. После обработки запроса с помощью объекта `request` пользователю отправляется ответная информация. С помощью метода `BinaryRead` объект `request` извлекает данные, передаваемые клиентом серверу, как часть запроса `POST`.

Объект `server` позволяет получить доступ к свойствам и методам Web-сервера. С помощью метода `CreateObject` можно создать экземпляр объекта `server`, `Execute` – выполняет файл `.asp`. Так же существует возможность сопоставления указанного виртуального пути с физическим путем – это делает метод `MapPath`. А `Transfer` передает всю информацию о текущем состоянии другому файлу `.asp` для обработки.

Объект `session` используется для хранения информации о пользовательских сеансах. Значения переменных объекта `session` сохраняются, даже когда пользователь переходит на другую страницу Web-приложения. Этот объект создается при организации пользователем сеанса и уничтожается при его завершении. Например, в нём можно сохранять регистрационную информацию каждого пользователя, посещающего сайт виртуального магазина. Эта информация остаётся доступной для всего Web-приложения даже при переходе пользователя на другие Web-страницы сайта. Объект `session` использует три метода: `Abandon` для уничтожения объект `session` и освобождения его ресурсов,

а Contents.Remove и Contents.RemoveAll для удаления одного элемента или всех элементов из семейства Contents соответственно.

Каждый из внутренних объектов ASP.NET обладает набором методов и коллекций для управления функциональными возможностями этого объекта. Назначение и возможности внутренних объектов технологий ASP и ASP.NET практически идентичны.

2.5 Описание программных модулей

Web-приложение состоит из следующих программных модулей: - модуль управления заказами клиентов;
- модуль администрирования Web-представительства. Ниже представлен PHP-класс формы вывода заказов <?php

```
$db = mysql_connect ("localhost", "root", "");
mysql_select_db ("mcd", $db);
$result = mysql_query ("SELECT id, fio, phone, mail, adress, tovar, sum from zakaz", $db);
$myrow = mysql_fetch_array ($result); do {
printf ("<form onSubmit = \"return checkform(this)\" action=\"final.php\"
method=\"post\">
    <table align='center' class='zak'> <tr
    bgcolor=\"#4FAB67\">
    <td style=\"text-align:center\"><strong> <p><u>id:</u>
    %s</p></strong></td>
    </tr>
    <tr bgcolor=\"#F7EEEE\"> <td>
    <p><u>Заказчик:</u> %s</p></td> </tr>
    <tr bgcolor=\"#4FAB67\"> <td>
    <p><u>Номер телефона:</u> %s</p></td> </tr>
    <tr bgcolor=\"#F7EEEE\"> <td>
    <p name=mail><u>e-mail:</u> %s</p><input type=\"submit\"
value=\"Отправить чек\"></td>
    </tr>
    <tr bgcolor=\"#4FAB67\"> <td>
    <p><u>Адрес:</u> %s</p></td> </tr>
    <tr bgcolor=\"#F7EEEE\"> <td>
    <p name=sum><u>Сумма заказа:</u> %s ?</p></td> </tr>
    <tr bgcolor=\"#4FAB67\"> <td><p
name=tovar><u>Заказ:</u> %s</p>
    </td>
    </tr>
    </table><br><br></form>", $myrow["id"], $myrow["fio"], $myrow["phone"],
$myrow["mail"], $myrow["adress"], $myrow["sum"], $myrow["tovar"]);
}
while ($myrow = mysql_fetch_array ($result)); ?>
```

2.6 Обоснование экономической эффективности разработки Web-

представительства ООО «Честер»

Для обоснования экономической эффективности разрабатываемого ИТ-решения предлагается методика сравнения себестоимости приобретенного программного продукта (ПП) (базовый вариант) и разработанного ПП (проектный вариант) программистом по договору (ИТ-аутсорсинг).

ИТ-аутсорсинг – предполагает делегирование внешней специализированной компании решение вопросов связанных с разработкой, внедрением и сопровождением информационных систем как целиком на уровне инфраструктуры предприятия (сопровождение оборудования или ПО), так и объемов работ, связанных с развитием и/или поддержкой функционирования отдельных участков системы (программирование, хостинг, тестирование и т. д.).

Проектируемый продукт представляет собой Web-сайт ООО «Честер» . Пользователями данной системы являются клиенты ресторана.

Стоимость приобретенного ПП (базовый вариант, $C_{ба}$) составила 25070 руб.

В процессе проектирования будут задействованы: менеджер, программист и администратор сайта.

В калькуляцию себестоимости разработки ПП включаются следующие статьи затрат:

- основная зарплата;
- дополнительная зарплата;
- социальные страховые взносы;
- прочие прямые расходы;
- накладные расходы.

Оплата труда представляет совокупность средств, выплаченных работникам в денежной и натуральной форме как за отработанное время, выполненную работу, так и в установленном законодательством порядке за неотработанное время.

Начисление основной заработной платы производится в зависимости от

принятых на предприятии форм оплаты труда. При повременной оплате труда основная заработная плата начисляется работникам за фактически отработанное время, а при сдельной за фактически выполненную работу.

Повременная форма оплаты труда находит применение при расчете заработной платы рабочих, служащих, специалистов и руководителей. При этой форме оплаты труда заработная плата рассчитывается исходя из месячного должностного оклада за проработанное время.

В статье «Дополнительная заработная плата» (ДЗП) планируются и учитываются выплаты, предусмотренные законодательством о труде или коллективными договорами за непроработанное на производстве (неявочное) время: оплата очередных и дополнительных отпусков, компенсация за неиспользованный отпуск, оплата льготных часов подросткам, оплата времени, связанного с выполнением государственных и общественных обязанностей и др. Она определяется в процентном отношении (10%) от основной заработной платы.

В статью «Накладные расходы» включаются расходы на управление и хозяйственное обслуживание. Величина накладных расходов определяется в процентах от основной и дополнительной заработной платы.

Накладные расходы составляют (НР) 40 % от фонда оплаты труда. Прочие прямые расходы (ППР) состоят из расходов на обслуживание ЭВМ, платы за потребляемую электроэнергию.

В случае ИТ-аутсорсинга отличающимися статьями затрат являются оплата труда стороннего программиста и связанные с ним расходы.

В компаниях, занимающихся разработкой и сопровождением информационных систем, принято оплачивать работу, исходя из количества, затраченного на разработку времени в часах и стоимости нормочаса специалиста.

Оценка часа работы сильно различается. Как правило, основные причины различий следующие:

- знание предметной области (рестораны быстрого питания);
- знание требуемой среды разработки (PHP+MySQL);

– составление технической документации.

Бухгалтерией ООО «Макдональдс» были произведены расчеты, в результате которых было принято решение установить часовую ставку стороннего программиста равной 500 руб. Результаты расчетов отражены в таблице 3.4.

Таблица 2.4 – Основная заработная плата исполнителей работ, проектный вариант

№	Этап	Исполнители		Время работы, час	Часовая ставка, руб	Размер заработной платы, руб.
		Должность	Кол-во			
1	Подготовительный	Менеджер	1	30	200	6000
2	Основной	Программист (по договору)	1	12	500	6000
3	Тестирование проекта	Администратор сайта	1	2	150	300
ИТОГО						12300

ДЗП рассчитывается только по штатным сотрудникам, что составляет $6300 \cdot 0.1 = 630$ руб.

ФОТ равен $12300 + 630 = 12930$ руб.

Сумма страховых взносов равна $12930 \cdot 0.3 = 3879$ руб., данные представлены в таблице 3.5.

Соответственно, уменьшатся ППР страховой компании: расходы на электроэнергию и машинное время:

ППР = $44 \cdot (3 + 0,35 \cdot 2,32) = 140$ руб

Итого ППР составляют $140 + 150 = 290$ руб.

НР равны $12930 \cdot 0,4 = 5172$ руб.

Таблица 2.5 – Расчет себестоимости внедрения проектного варианта

Статьи затрат	Сумма затрат (руб.)	Удельный вес, %
1. Основная зарплата	12300	52
2. Дополнительная зарплата (10 % от основной)	630	2
3. Отчисления в ПФ РФ (22% от ФОТ)	2845	12
4. Отчисления в ФСС РФ (2,9% от ФОТ)	375	1
5. Отчисления в ОМС (5.1% от ФОТ)	659	2
4. Прочие прямые затраты	290	1
5. Накладные расходы	5172	30
ИТОГО	22271	100

Таким образом, себестоимость разработки программы на условиях ИТ-аутсорсинга равна $С_{пр} = 22271$ руб.

Формируем таблицу показателей эффективности (таблица 2.6 и рисунок 2.9).

Таблица 2.6 - Показатели эффективности от внедрения проекта
автоматизации

	Затраты		Абсолютное изменение затрат	Коэфф-т изменения затрат	Индекс изменени я затрат
	Базовый вариант	Проектный вариант			
Стоимость	$C_{\text{баз}}$ (руб.)	$C_{\text{пр}}$ (руб.)	$\Delta C = C_{\text{баз}} - C_{\text{пр}}$ (руб.)	$K_C = \Delta C / C_{\text{баз}} \times$ 100%	$Y_C = C_{\text{баз}} /$ $C_{\text{пр}}$
	25070	22271	2799	11,2 %	1,13

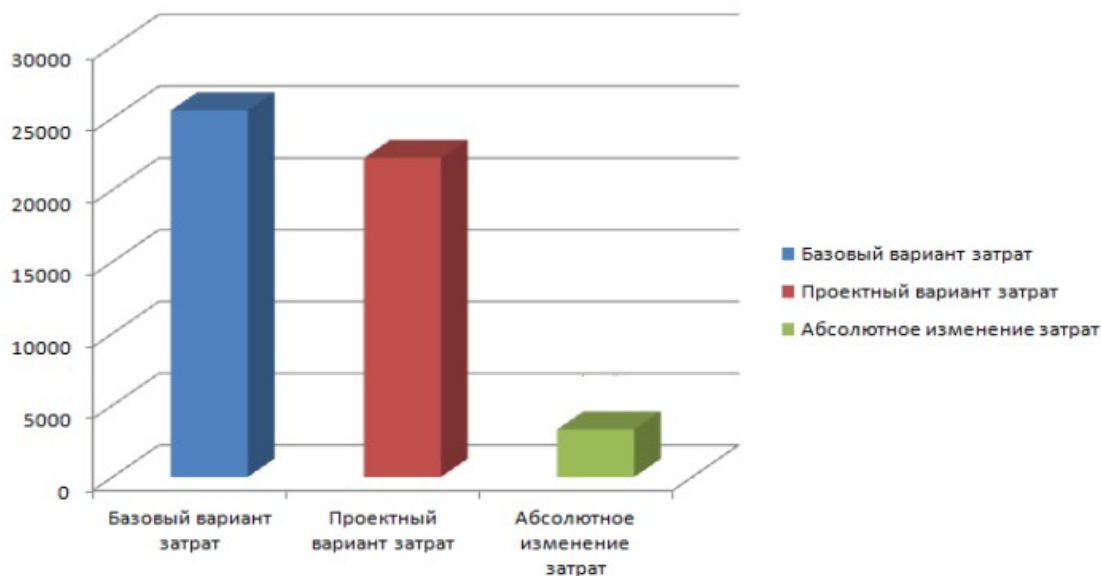


Рисунок 2.10 – Графики показателей эффективности базового и проектного вариантов разработки Web-представительства

Помимо рассмотренных показателей целесообразно также рассчитать срок окупаемости затрат на внедрение проекта машинной обработки информации ($T_{ок}$):

Срок окупаемости затрат на внедрение проекта машинной обработки информации ($T_{ок}$):

$$T_{ок} = K_{п} / \Delta C \text{ (мес.)},$$

где $K_{п}$ - затраты на создание проекта машинной обработки информации (проектирование и внедрение).

Единовременные затраты в $K_{п}$ сфере использования в данном случае складываются из затрат на проектирование Web-представительства.

Следовательно, срок окупаемости Web-представительства равен:

$$T_{ок} = 22271 / 2799 = 8 \text{ мес.}$$

Сравнение себестоимостей базового и проектного вариантов разработки Web-сайта ООО «Честер» подтвердило целесообразность разработки на условиях ИТ-аутсорсинга, в рамках которой выполнена курсовой проект.

Заключение

Курсовая работа посвящена актуальной проблеме разработки Web-сайта ООО «Честер». В процессе выполнения курсовой работы достигнуты следующие результаты:

- 1) произведен анализ предметной области. На основе структурного подхода и методологий IDE0 и DFD разработана концептуальная модель Web-представительства ООО «Честер»;
- 2) сформулированы требования к Web-сайту ООО «Честер»;
- 3) произведен анализ известных ИТ-решений, по результатам которого принято решение о разработки нового Web-сайту ООО «Честер»;
- 4) на основе методологии объектно-ориентированного анализа и языка UML разработана логическая модель Web-сайта;
- 5) на основе методологии IDEF1X разработана логическая модель данных Web-сайта;
- 6) с помощью технологии ASP.NET разработано программное приложение Web-сайта ООО «Честер» ;
- 7) обоснована экономическая эффективность разработки Web-сайта.

Таким образом поставленные цели достигнуты в полном объеме.

Список используемой литературы

Нормативно-правовые акты:

1. ГОСТ 34.601-90. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
2. ГОСТ 34.320-96. Информационная технология. Система стандартов по базам данных. Концепции и терминология для концептуальной схемы и информационной базы.
3. ГОСТ Р ИСО/МЭК 12207-99. Информационная технология. Процессы жизненного цикла программных средств.

Учебники и учебные пособия:

4. Гагарина Л. Г., Киселев Д. В., Федотова Е. Л. Разработка и эксплуатация автоматизированных информационных систем М.: ИД «ФОРУМ», ИНФРА-М, 2012. – 384 с.
5. Голицына, О. Л. Системы управления базами данных : учеб. пособие / О. Л. Голицына, Т. Л. Партыка, И. И. Попов. - Гриф МО. – М. : ФОРУМ -ИНФРА-М, 2011. - 431 с.
6. Дудина, И.П. Рекомендации по выполнению выпускной квалификационной работы бакалавра по направлению подготовки «Прикладная информатика»: учеб.- метод. пособие / И.П. Дудина, О.М. Гущина, С.В. Мкртычев. – Тольятти: Изд-во ТГУ, 2013. – 59 с.
7. Золотов, С. Ю. Проектирование информационных систем : учеб. пособие / С. Ю. Золотов ; Томский гос. ун-т систем управления и радиоэлектроники. - Томск : Эль Контент, 2013. - 86 с.
8. Карпова, И. П. Базы данных : курс лекций и материалы для практ. занятий : учеб. пособие для студентов техн. фак. / И. П. Карпова. – СПб. : Питер, 2013. - 240 с.
9. Малышев С. Л. Основы интернет-экономики : учеб. пособие / С. Л. Малышев. – М. : Евразийский открытый институт, 2011. - 118 с.

10. Реинжиниринг бизнес-процессов : учеб. пособие / А. О. Блинов [и др.] ; под ред. А. О. Блинова. - Москва : ЮНИТИ-ДАНА, 2012. - 341 с.
11. Рудинский, И. Д. Технология проектирования автоматизированных систем обработки информации и управления : учеб. пособие / И. Д. Рудинский. – М. : Горячая линия - Телеком, 2011. - 304 с.
12. Шелухин, О. И. Моделирование информационных систем: учеб. пособие. 004 / О. И. Шелухин. - 2-е изд., перераб. и доп. – М. : Горячая линия - Телеком, 2012. - 516 с.
13. Юрасов, А. В. Интернет-маркетинг : учебное пособие / А. В. Юрасов, А. В. Иванов ; под ред. А. В. Юрасова. – М. : Горячая линия - Телеком, 2012. - 246 с.

Фрагменты программного кода приложения Web-сайта

**** Выбор позиции меню**

```
<?php
header('Content-Type: text/html; charset=utf-8');
setlocale(LC_ALL, 'ru_RU.65001', 'rus_RUS.65001', 'Russian_Russia. 65001', 'russian');
session_start();
if (isset($_SESSION['talon']) == "" ) {
include('bd.php');
$result = mysql_query("SELECT counter FROM posetiteli", $conn); if (!$result) {echo
"zaprosh na viborcu ne proshol."; mysql_error();} $x = mysql_fetch_array($result);
$_SESSION['talon'] = $x["counter"]+1;
$result = mysql_query("UPDATE posetiteli SET counter = counter + 1", $conn);
if (!$result) {echo "zaprosh na viborcu ne proshol."; mysql_error();}
mysql_close($conn);
} ?>
<?php
// Connect to database
include('bd.php');
//Формирование оператора SQL SELECT
$sqlCart = mysql_query("SELECT id_tovara, colichestvo FROM vibranie_tovari
WHERE talon = '$_SESSION[talon]'", $conn);
//Цикл по множеству записей и вывод необходимых записей $OrderTotal=0;
```

```

while($row = mysql_fetch_array($sqlCart))
{
//Присваивание записей $colichestvo =
$row["colichestvo"]; $id_tovara =
$row["id_tovara"];
//Формирование оператора SQL SELECT
$sqlProd = mysql_query("SELECT name, prise FROM tovari WHERE id = '$id_tovara'", $conn);
//Выполнение оператора SQL и создание множества записей $row2 =
mysql_fetch_array($sqlProd);
//Присваивание записей $talon =
$_SESSION['talon']; $name =
$row2["name"]; $prise = $row2["prise"];
$resultat = ($prise*$colichestvo); $OrderTotal =
$OrderTotal + $resultat;
}
mysql_close($conn); ?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html> <head>
<title>САНДВИЧИ</title>
<meta http-equiv="Content-Style-Type" content="text/css">
<link rel="stylesheet" type="text/css" href="css/reset.css" media="screen" /> <!-- Main CSS File --
><link rel="stylesheet" type="text/css"
href="css/style.css" media="screen" />
<!-- These styles are only for demo--> <style
type="text/css">

```



```

#container {width:960px;margin-left: auto;margin-right: auto;padding: 0;}
.clear {clear: both;margin: 0;padding: 0;} p {margin:
1.6em 0;line-height: 1.6em;}
h2 {font-size: 20px;line-height: 20px;margin: 22px 0 18px 0;} </style>
</head>
<body background="img/1.jpg"> <div
id="menu_div">
<div id="navigation"> <div id="menu">
<ul id="nav">
<li><a href="index.php">Главная</a> <!-- END Home Item -->
<li><a href="about.php">О ресторане</a> <li><a href="map.php">Найти на карте</a>
<li><a href="menu.php">Доставка</a><ul> <li><a
href="corzina.php">Оформить заказ</a></li>
</ul></li> <li><a
href="https://www.mcdonalds.ru/">Официальная страница</a></li> </ul><!--
#nav END-->
<!-- Search Form --> <form class="searchform" >
<textarea disabled>Общая сумма заказа:<?php echo number_format($OrderTotal,2) ?> ?
</textarea>
</form>
<!-- / Search Form --> </div><!-- #menu END-->
</div><!-- #navigation END-->

```

```
</div><!-- #menu_div END--><span class="hidden"><span
class="menu_item"><a class="mi_prod_img"
href="https://www.mcdonalds.ru/main-menu/sandwiches/big-
mac#id=9"></a></span></span>
<table style="margin-left:50px" style="margin-top:" align="center" width="1075"
height="694" border="2" cellspacing="20">
<span class="hidden"><span class="menu_item"><a class="mi_prod_img"
href="https://www.mcdonalds.ru/main-menu/sandwiches/big-mac#id=9"></a></span></span>
<tbody> <tr>
<td width="319">&nbsp;</td> <td
width="320">&nbsp;</td> <td
width="340">&nbsp;</td> </tr>
<tr>
<td height="357">&nbsp;</td>
<td>&nbsp;</td> <td>&nbsp;</td>
</tr> </tbody>
<a class="mi_prod_img"
href="https://www.mcdonalds.ru/main-menu/sandwiches/hamburger#id=10"><span
class="hidden"></span></a><span class="hidden"><span class="menu_item"><a
class="mi_prod_img" href="https://www.mcdonalds.ru/main-menu/sandwiches/big-mac#id=9"><a
class="mi_prod_img" href="https://www.mcdonalds.ru/main-menu/sandwiches/big-mac#id=9"><a
class="mi_prod_img" href="https://www.mcdonalds.ru/main-menu/sandwiches/royal-de-
luxe#id=18"></a></a></span></span>
<!--Гамбьпреп-->
```

```
<form action="zapisi_burger.php?idtovara=1" method="POST"><span
class="menu_item"><a class="mi_prod_img" href="/burger/1.jpg"><input type="image"
src="/burger/1.jpg"></a></span></form>
<!--Двойной чизбургер-->
<form action="zapisi_burger.php?idtovara=2" method="POST"><span
class="menu_item"><a class="mi_prod_img" href="/burger/2.jpg"><input type="image"
src="/burger/2.jpg"></a></span></form>
<!--Чикен Бекон-->
<form action="zapisi_burger.php?idtovara=3" method="POST"><span
class="menu_item"><a class="mi_prod_img" href="/burger/3.jpg"><input type="image"
src="/burger/3.jpg"></a></span></form>
<!--Роял Де Люкс-->
<form action="zapisi_burger.php?idtovara=4" method="POST"><span
class="menu_item"><a class="mi_prod_img" href="/burger/4.jpg"><input type="image"
src="/burger/4.jpg"></a></span></form>
<!--Роял Чизбургер-->
<form action="zapisi_burger.php?idtovara=5" method="POST"><span
class="menu_item"><a class="mi_prod_img" href="/burger/5.jpg"><input type="image"
src="/burger/5.jpg"></a></span></form>
<!--Чизбургер-->
<form action="zapisi_burger.php?idtovara=6" method="POST"><span
class="menu_item"><a class="mi_prod_img" href="/burger/6.jpg"><input type="image"
src="/burger/6.jpg"></a></span></form>
<!--Биг Мак-->
<form action="zapisi_burger.php?idtovara=7" method="POST"><span
class="menu_item"><a class="mi_prod_img" href="/burger/7.jpg"><input type="image"
src="/burger/7.jpg"></a></span></form>
<!--Чикенбургер-->
```

```
<form action="zapisi_burger.php?idtovara=8" method="POST"><span
class="menu_item"><a class="mi_prod_img" href="/burger/8.jpg"><input type="image"
src="/burger/8.jpg"></a></span></form>
<!--Биг Тейсти-->
<form action="zapisi_burger.php?idtovara=9" method="POST"><span
class="menu_item"><a class="mi_prod_img" href="/burger/9.jpg"><input type="image"
src="/burger/9.jpg"></a></span></form>
<!--Фрешбургер с сыром-->
<form action="zapisi_burger.php?idtovara=10" method="POST"><span
class="menu_item"><a class="mi_prod_img" href="/burger/10.jpg"><input type="image"
src="/burger/10.jpg"></a></span></form>
<!--Макчикен-->
<form action="zapisi_burger.php?idtovara=11" method="POST"><span
class="menu_item"><a class="mi_prod_img" href="/burger/11.jpg"><input type="image"
src="/burger/11.jpg"></a></span></form>
<!--Филе-о-Фиш-->
<form action="zapisi_burger.php?idtovara=12" method="POST"><span
class="menu_item"><a class="mi_prod_img" href="/burger/12.jpg"><input type="image"
src="/burger/12.jpg"></a></span></form>
</body> </html>
```