

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Кафедра Информационные системы

«Допущен к защите»  
Заведующий кафедрой \_\_\_\_\_

(Ф.И.О., ученая степень, звание)

« \_\_\_\_\_ » 20\_\_ г.

(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка автоматизированной информационной системы «Barpetit»

Специальность 5В070300 - «Информационные системы»

Выполнил (а) Туктаров В. П. ИС-10-2  
(Фамилия и инициалы) группа

Научный руководитель к.т.н., доцент Ни. А. П.  
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Семашева А. Ч., к.т.н., доцент  
(Фамилия и инициалы, ученая степень, звание)  
А. Ч. « 4 » апрель 2014 г.  
(подпись)

по безопасности жизнедеятельности:

Башкирская А. С., ст. преп.  
(Фамилия и инициалы, ученая степень, звание)  
А. С. « 22 » апрель 2014 г.  
(подпись)

по применению вычислительной техники:

Касымбаева Б. К., к.т.н., ст. преподаватель  
(Фамилия и инициалы, ученая степень, звание)  
Б. К. « 03 » 06 2014 г.  
(подпись)

Нормоконтролер: Ни А. П., к.т.н., доцент  
(Фамилия и инициалы, ученая степень, звание)  
А. П. « 05 » 06 2014 г.  
(подпись)

Рецензент: \_\_\_\_\_  
(Фамилия и инициалы, ученая степень, звание)  
« \_\_\_\_\_ » 20\_\_ г.  
(подпись)

Алматы 2014 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет «Информационные технологии»  
Специальность «68070300 - Информационные системы»  
Кафедра «Информационные системы»

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Жоктаров Бауржан Жоктарулы  
(фамилия, имя, отчество)

Тема проекта Разработка автоматизированной информационной системы «Varretit»

утверждена приказом ректора №     от «   » сентября 20    г.

Срок сдачи законченной работы «   »     20    г.

Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

Исходные данные: Задача к дипломному проекту. Требуемые показатели результатов проектирования: Модель АИС «Varretit», ER-диаграмма базы данных, программная реализация приложений, база данных АИС «Varretit». Исходные данные объекта: результаты обследования предметной области.

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

1. Анализ и исследование информационных технологий в розничном бизнесе
2. Проектирование и разработка АИС «Varretit»
3. Разработка программного обеспечения АИС «Varretit»
4. Технико-экономическое обоснование проекта
5. Анализ условий труда

Перечень графического материала (с точным указанием обязательных чертежей)

1. Диаграмма цен/ценностей
2. Диаграмма видов деятельности
3. Диаграмма последовательности
4. Диаграмма составных
5. Диаграмма компонентов
6. Диаграмма развертывания
7. Логическая схема базы данных
8. Физическая схема базы данных

#### Рекомендуемая основная литература

1. Карагеровский В. ИКТ в ресторанном бизнесе - текущая потребность или опережающие времена [Электронный ресурс]. - Режим доступа: [http://www.news.ru/reviews/free/trade2008/articles/restaurant\\_3.shtml](http://www.news.ru/reviews/free/trade2008/articles/restaurant_3.shtml)
2. Диаграмма ИМЭ [Электронный ресурс]. - Режим доступа: <http://www.intuit.ru/studies/courses>
3. Бекмурзаева А.И. Методические указания к выполнению экономической части дипломной работы для бакалавров специальности 58070800 - ИИ - формационные системы - Алматы: АУЭС; 2013. - 24 с.
4. Хакимжанов Т.И. Методические указания к выполнению раздела в дипломных проектах (для студентов бакалавриата всех специальностей) - Алматы: АУЭС; 2002. - 29 с.

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
Экспл. часть	Бекмурзаева А.И		
БТД	Бекмурзаева А.И		
Осн. часть	Жан А.Т.	05.06.14	

**Г Р А Ф И К**  
подготовки дипломного проекта

№ п/п	Наименование разделов, перечень разрабатываемых вопросов	Сроки представления руководителю	Примечание
1	ИТ в ресторанном бизнесе. Анализ применяемых информационных технологий в ресторанном бизнесе и модели автоматизации бизнес-процесса кафе «Bon Appetit». Обзор международных и внутренних средств.	01.11.13 - 01.01.14	
2	Проектирование и разработка АИС «Bon Appetit». Системный анализ, разработка ИМЛ-диаграммы и баз данных АИС «Bon Appetit».	01.01.14 - 01.04.14	
3	Разработка программного обеспечения АИС «Bon Appetit». Написание и условия формирования программы. Разработка интерфейса АИС «Bon Appetit».	01.04.14 - 01.06.14	
4	Технико-экономическое обоснование проекта.	04.04.14 - 04.05.14	
б.	Безопасности компьютерных сетей.	22.03.14 - 22.04.14	

Дата выдачи задания «    »    20    г.

Заведующий кафедрой \_\_\_\_\_  
(подпись) (Фамилия и инициалы)

Руководитель \_\_\_\_\_  
(подпись) (Фамилия и инициалы)

Задание принял к исполнению студент \_\_\_\_\_  
(подпись) (Фамилия и инициалы)

## **датпа**

Дипломдық жобада автоматтандырылған ақпараттық «Bappetit» деген жүйе жасалған.

Дипломдық жобада бүгінгі күнде мейрамханалық бизнесінде қолданылатын ақпараттық технологиялардың талдауы орындалған. Сонымен қатар «Bappetit» жүйесіне арналған UML-диаграммалар мен дерекқор жасалған.

«Bappetit» жүйесі «BonAppetit» кафесінің бизнес-процесстерін автоматтандыру мақсатпен жасалған.

## **Аннотация**

В дипломном проекте разработана автоматизированная информационная система «Bappetit». Осуществлен анализ существующих информационных технологий в ресторанном бизнесе, разработаны UML диаграммы и база данных АИС «Bappetit».

АИС «Bappetit» разработана с целью автоматизации бизнес-процессов кафе «BonAppetit».

## **Annotation**

In the graduation project was developed the automated information system of "Bappetit". The analysis of existing information technologies in restaurant business was realized, UML charts and database AIS "Bappetit" were developed.

AIS "Bappetit" is developed for the purpose of automation of business processes of "BonAppetit" cafe.

## Содержание

Введение	9
1 Информационные технологии в ресторанном бизнесе	10
1.1 Применение информационных технологий в ресторанном бизнесе	10
1.2 Проблемы автоматизации бизнес-процесса кафе «BonAppetit»	15
1.3 Обзор используемых программных и инструментальных средств	15
2 Проектирование и разработка АИС «Vappetit»	20
2.1 Системный анализ объекта исследования	20
2.2 Разработка UML-диаграмм АИС «Vappetit»	21
2.3 Разработка базы данных АИС «Vappetit»	28
3 Разработка программного обеспечения АИС «Vappetit»	30
3.1 Назначение и условия выполнения программы	30
3.2 Разработка интерфейса АИС «Vappetit»	34
4 Технико-экономическое обоснование проекта	45
4.1 Технико-экономическое обоснование	45
4.2 Расчет трудоемкости разработки АИС «Vappetit»	46
4.3 Расчет затрат на разработку АИС «Vappetit»	47
4.4 Определение возможной (договорной) цены АИС «Vappetit»	51
4.5 Расчет срока окупаемости АИС «Vappetit»	51
4.6 Оценка социально - экономических результатов функционирования АИС «Vappetit»	52
5 Безопасность жизнедеятельности	53
5.1 Анализ условий труда в рабочем помещении	53
5.2 Расчет системы кондиционирования	56
Заключение	63
Список литературы	64
Приложение А Техническое задание	65
Приложение Б Листинг программы	82

## Введение

В любом кафе, будь то многоэтажный комплекс или кафе на 5 столиков, происходят одни и те же рабочие процессы. Автоматизация процессов призвана повысить прозрачность работы и, как следствие, прибыль заведения. В случае с небольшим кафе, контроль рабочих процессов позволяет сэкономить относительно большую часть прибыли, а организация автоматизации проще и дешевле.

Общая оценка рынка благоприятна для ПП подобного рода. Новые технологии расширяют функционал, сферы их применения и увеличивают число постоянных клиентов, сокращают злоупотребление среди рабочего персонала, облегчают работу заведующего производством.

Введение такой системы очень оправдано, в первую очередь, для небольших заведений, в которых недобросовестность персонала может очень чувствительно сказываться на доходе, который и так не слишком велик. И подобные заведения просто очень нуждаются в очень эффективных инструментах управления. Если правильно организовать дело с самого начала, то даже небольшое заведение может очень быстро окупиться и начать давать хороший доход.

Программный продукт (АИС Varpetit) данной дипломной работы представляет собой управленческую бэк-офисную часть решения для автоматизации кафе и заведений общественного питания. Данные решения позволяют вести товарный учет, рассчитывать калькуляцию и редактировать меню, получать различную аналитическую отчетность, ускорять взаимодействие во фронт-офисной части кафе и ресторанов.

Создание АИС Varpetit преследует достижение следующих основных целей:

- решение регламентных задач финансового учета заказов и материалов;
- снижение временных затрат на обработку заказов;
- получение отчетов о выполненных операциях, используя созданную базу данных;
- автоматическое ведение и обновление количественно-суммовой статистики;
- повышение эффективности и оперативности работы сотрудников кафе.

## **1 Информационные технологии в ресторанном бизнесе**

### **1.1 Применение информационных технологий в ресторанном бизнесе**

Ресторанный бизнес одна из перспективно развивающихся отраслей индустрии гостеприимства. Люди приходят в ресторан не только отведать изысканные блюда, но и отдохнуть, получить эстетическое удовольствие и просто пообщаться друг с другом. Достижение успеха в развитии любого бизнеса, в том числе и ресторанного, зависит от множества факторов, одним из которых является использование информационных технологий.

Современные информационные компьютерные технологии, разработанные специально для предприятий общественного питания, позволяют значительно упростить, оптимизировать и ускорить целый ряд рутинных, специфических для этого бизнеса операций. В настоящее время для автоматизации предприятий общественного питания используют следующие системы автоматизации: «R-Keeper», «IikoChain», «1С-Рарус», «UNISYSTEM Ресторан», «X-КАФЕ», «Microinvest Склад Pro» и другие.

Современный ресторан кафе, бар, клуб, отель – это, прежде всего, красивый и комфортный способ времяпрепровождения, один из основных элементов досуга. Организация подобного процесса крайне сложна по своему содержанию и наполнению и требует: соблюдения санитарных и технологических норм, контроля за стилем и культурой поведения обслуживающего персонала, за учетным процессом, анализа транзакций, учета поступления продуктов, формирования стоимости блюд и полуфабрикатов, процедур списания продуктов. Требование автоматизации всех этих процессов вытекает, прежде всего, из необходимости учёта большого количества деталей.

Удобство автоматизации и информатизации процессов на предприятии общественного питания очевидно не только с точки зрения «ведения дел», но и с позиций клиентов, так как информационные системы позволяют более оперативно осуществлять расчеты с посетителями, очередностью обслуживания, обеспеченностью предлагаемого меню всеми необходимыми ингредиентами[1].

Например, внедрение систем оплаты по пластиковым картам и депозитно-дисконтных систем, позволяет резко расширить возможности заведения ресторанного хозяйства, организуя совместные акции с ведущими банками. Системы не просто поддерживают безналичную форму оплаты, но и ведут учет постоянных и корпоративных клиентов, позволяют управлять системой скидок и бонусов, поддерживать удаленный доступ клиентов для формирования заявок и заказов. Большой потенциал получают всякого рода программы по повышению лояльности клиентов (скидки, бонусы, специальные предложения и т.д.), формирование постоянной клиентской базы.



Целью автоматизации ресторанного бизнеса является повышение эффективности управления предприятием питания, ускорение скорости и качества обслуживания гостей, минимизация злоупотреблений персонала.

Программы для автоматизации ресторана должны быть многофункциональными системами для повышения прибыльности и снижения издержек предприятия, легкими во внедрении и простыми в обслуживании[1].

Автоматизация ресторана с помощью программ компании «1С-Рарус» позволяет оптимально сочетать все функции, необходимые для эффективного управления рестораном, предоставляя высокую надежность и качество.

Программа «R-Keer» предназначена для автоматизации продаж, обслуживания клиентов, работы кухни и баров, накопления и анализа данных по продажам и сервису за различные промежутки времени на предприятиях общественного питания и центрах индустрии развлечений. Накопленные данные используются в системах складского учета и бухгалтерского учета.

Продукт "1С:Предприятие 8. Ресторан" предназначен для автоматизации процессов обслуживания посетителей на предприятиях индустрии питания и гостеприимства. С помощью этого решения могут быть автоматизированы одиночные и сетевые предприятия любых форматов и концепций: рестораны, кафе, бары, столовые, гостинично-ресторанные комплексы, развлекательные центры и другие предприятия питания[2].

Программный продукт «IikoChain» - это решение для успешного управления как отдельным рестораном или кафе, так и ресторанной сетью и корпорацией в целом. Данная программа позволяет осуществлять отчетность в разрезе по ресторанам и общую отчетность ресторанной сети по продажам, производству, персоналу и финансам, а также управление остатками и складскими запасами на уровне всех подразделений.

Программный комплекс «UNISYSTEM Ресторан» осуществляет централизованный товарный и финансовый учет предприятия, ориентированный на глобальный контроль движения товара и денежных средств. Система мониторинга торгового зала позволяет отслеживать все операции по обслуживанию клиентов официантами.

Еще один пионер для автоматизации ресторанов (16 лет на рынке). Производитель также обещает возможность управлять ресторанами любых размеров. В систему TillyPad входят следующие механизмы:

– TillyKey – система для распределения прав доступа на основе занимаемой должности. Выделяет полномочия для работы в системе в зависимости от обязанностей. Действительно, официанту совершенно не нужен доступ к управлению персоналом. Такого рода механизмы необходимы любой системе автоматизации.

– TillyLing – механизм, позволяющий использовать мультязычный интерфейс для всех работников. Использование нескольких языков частично снимает языковой барьер, позволяя нанимать работников, которые не общаются непосредственно с посетителями (поваров, бухгалтеров и так далее), вне зависимости от их владения языком.

– TillyLog – как ясно из названия данная система обеспечивает сохранение всех изменений, произведенных в системе, для последующего анализа. Благодаря таким механизмам становится возможным прогнозирование работы персонала и другие, важные для бизнеса процессы.

– TillySync – механизм гибкой синхронизации, который позволяет территориально расширять организацию без потерь в производительности. Особенно актуально для сетей ресторанов.

TillyStock – механизм для управления складом, поставками и подобным. Механизм разработан с учетом специфики ресторанного склада.

TillyVision – система видеонаблюдения для ресторана. Позволяет использовать произвольное количество камер и мониторов, выдавать изображение с нескольких камер и мониторов, а также удаленно управлять терминалами.

Что касается аппаратного обеспечения, система предлагает использовать:

- POS-терминалы;
- КПК;
- принтеры;
- фискальные регистраторы;
- программируемые клавиатуры;
- считыватели магнитных карт, штрих кодов и подобного;
- системы вызова официантов;
- устройства управления электропотребления;
- и другое оборудование.

Благодаря возможности использования TillyPad на всех уровнях работы (как фронт-офис, так и бэк-офис), исключается ошибки интеграции с другими системами по управлению бизнесом. Впрочем, в системе отсутствует управление бухгалтером и документооборотом, а также событийный видеоконтроль. Без событийного видеоконтроля становится достаточно трудно вычислить проблему с воровством персонала, но это все же не самый большой недостаток.

ACU Z-Cash Restaurant Suite – это бесплатная система с открытым исходным кодом. Состоит из трех частей: Front-, Back- и Product-Office. Полностью описать все достоинства и недостатки данной системы не представляется возможным из-за открытого исходного кода, который позволяет дорабатывать систему под себя в неограниченных масштабах. Основной недостаток системы ее вечная «недоработанность» — из-за того, что можно добавить все что угодно, всегда будет что-то, что нужно еще добавить. Этот нюанс заставит хозяина ресторана принять на работу программистов, для доведения системы до идеала. Но это же и ее достоинство – открытый исходный код позволит встроить в систему все, что угодно, начиная от действительно необходимых функций, заканчивая дополнительными приятными частями, без которых можно было бы и обойтись.

На данный момент в системе есть: Front Office:

- заказа блюд и напитков;
- изменение заказанных блюд и напитков;
- печать заказов на кухне или вывод их на терминал;
- счета;
- фискальные чеки;
- закрытие счетов;
- скидки по клиентским картам;
- оплата клиентской картой;
- учет рабочих часов;
- и так далее.

Back Office:

- просмотр столов в режиме реального времени;
- работа с выручкой (реализованная и нереализованная выручка) в реальном времени;
- просмотр состояния склада в реальном времени;
- просмотр архива смен с возможностью просмотра каждого счета;
- управление пользователями (добавление, редактирование и удаление пользователей);
- управление правами (добавление должности, редактирование прав доступа);
- дисконтные программы (добавление, редактирование и удаление дисконтных программ);
- работа с меню (добавление, редактирование, удаление и временная деактивация блюд, напитков и каталогов и так далее);
- работа со складскими актами (транзакции актов, редактирование акта до его закрытия, просмотр всех актов и так далее);
- и многое другое.

Product Office:

- отображение заказа (различные возможности отображения);
- информация о возможных стоп-листах;
- информация о времени, затраченном на приготовление блюд и напитков;
- справочник карт технологических процессов;
- мониторинг времени приготовления блюд и напитков.

Бесплатность системы с такими возможностями – несомненный плюс, но система автоматизации состоит не только из программного комплекса, но и из аппаратного, который бесплатным быть не может ни при каких обстоятельствах. С другой стороны Z-Cash не привязан ни к какой платформе (в отличие от R-Keerer, который привязан к планшетах фирмы Apple)[2].

Астор Лайт позиционируется авторами программы как система учета для небольших и средних ресторанов. Первая в списке обозреваемых программ, которая не заявляет о своей универсальности и масштабируемости, что сразу можно записать в плюсы системе. Программный комплекс АСТОР

предназначен для автоматизации как фронт-офиса, так и бэк-офиса. В данную систему входят приложения для:

- кассиров;
- менеджеров зала;
- калькуляции;
- управляющего;
- заведующего производством;
- кладовщика;
- менеджера по закупкам;
- официантов.

Для каждого из работников существует учетная запись с ограничением прав, благодаря которой, каждый занимается своим делом и не может случайно или намеренно испортить работу остальных. Функционально система фронт-офиса не представляет из себя ничего необычного, а бэк-офис дополнительно предоставляет интерфейс для множественных закупок продуктов, взаиморасчетов внутри и вне ресторана. Как и предполагают разработчики, АСТОР лучше всего подходит для небольших ресторанов, которые пользуются услугами нескольких поставщиков. А возможно подойдет и сети небольших кафе и фаст-фудов.

Система под названием «Рестораторъ» от компании РСТъ – позиционируется как «недорогое решение для автоматизации ресторана». Особенность данного комплекса программного обеспечения в его модульности. То есть, заказчик системы сам выбирает какие модули программы ему необходимы, а остальные либо не покупает, либо покупает позже, когда ресторан начинает расти и требовать дополнительных функциональных возможностей автоматизации. Плюсы такого подхода очевидны и в наше время внедряются повсеместно (так называемые «облачные технологии» как раз и позволяют платить только за то, что вам необходимо). Фронт- и бэк-офис программы распространяются бесплатно, а оплачивать необходимо лишь дополнения, такие как:

- модуль для работы склада;
- модуль для работы клуба;
- и так далее.

Единственное ограничение бесплатной лицензии – количество задействованных рабочих мест (в бесплатной всего 2 рабочих места, что годится только для небольшого кафе). В зависимости от необходимых модулей и количества рабочих мест – изменяется и стоимость Ресторатора. В отличие от Z-Cash, которая также модульна и полностью бесплатна, Рестораторъ предлагает готовое решение, которое не будет нужно будет дорабатывать самостоятельно. Компания РСТъ также предлагает и аппаратную часть для своего комплекса, что позволяет использовать ее, что называется «из коробки» без сложностей с интеграцией с нестандартным оборудованием. Система автоматизации ресторанов Рестораторъ отлично подойдет тем, кто начинает бизнес практически с нуля, так как позволит

стартовать систему с относительно небольшими затратами, чтобы затем расширить функциональность системы до полной автоматизации[2].

Автоматизация ресторанного бизнеса позволяет исключить потери прибыли по причине человеческого фактора, увеличить скорость обслуживания клиентов и даже выработать маркетинговую стратегию бизнеса.

Таким образом, только при формировании грамотно разработанной концепции и последовательного комплексного внедрения всех составляющих ресторанного бизнеса, одной из которых являются информационные технологии, гарантирован успех в развитии деятельности предприятий ресторанного хозяйства.

## **1.2 Проблемы автоматизации бизнес-процесса кафе «BonAppetit»**

Приоритетными задачами кафе «BonAppetit» являются привлечение новых клиентов, и поддержание лояльности уже существующих клиентов, и получение, обработка и предоставление оперативной информации о заказах. Очень важными являются оптимизация закупок, управление запасами материала, снижение уровня «нелояльности» персонала, повышение прозрачности и эффективности бизнеса, систематизации всех процессов и управления ими.

Можно выделить основные проблемы:

- борьба с воровством и махинациями со стороны персонала;
- повышение эффективности работы заведения за счет оперативного контроля заказов, и, как следствие, снижение издержек;
- улучшение качества обслуживания клиентов;
- упрощение работы администратора.

В связи с возникшими проблемами было решено создать автоматизированную информационную систему «Vappetit», которая позволит оперативно производить все процессы, необходимые для корректного ведения товаручета и обработки заказов, а также предоставлять различную аналитическую отчетность и статистику по продуктам и заказам, обеспечивать возможность изменения дисконтной политики и ценовых позиций в меню.

## **1.3 Обзор используемых программных и инструментальных средств**

На сегодняшний день существует множество различных программных и инструментальных средств, с помощью которых можно создать информационные системы.

На современном этапе рынок информационных технологий предлагает большое количество инструментальных средств разработки информационных систем, позволяющих оперативно решать проблемы управления кадрами, повышать эффективность бизнес-процессов в организациях любого уровня

и различных направлений деятельности. Разнообразие таких программных продуктов позволяет выбрать оптимальное инструментальное средство, позволяющее удовлетворить потребности конечного пользователя в требуемом объеме.

Кроме того, при создании баз данных важным является выбор такого инструментального средства разработки, который бы удовлетворял такие потребности конечных пользователей как: актуальность информации для организации; производительность БД, обеспечивающая получение требуемых данных за короткое время; гибкость и возможность подстройки под пользователя; масштабируемость при реорганизации и расширении предметной области; кроссплатформенность; целостность базы данных; удобство использования.

В данном дипломном проекте была использована MySQL. MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, 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.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц[9].

Преимущества применения СУБД MySQL описаны ниже:

– сокращение времени, необходимого для ведения записей. В случае использования СУБД не требуется много времени на просмотр всей картотеки, чтобы добавить новую запись. Вы просто вводите ее в систему, не заботясь о месте размещения;

– сокращение времени, необходимого для поиска записей. При поиске данных в СУБД нет необходимости последовательно просматривать все записи, чтобы найти интересующую. Предположим, что вы работаете в стоматологическом кабинете и вам надо разослать приглашения всем пациентам, которые забыли пройти ежегодный профилактический осмотр. Для этого достаточно сделать запрос к информационной системе. Конечно, это происходит не так, как при обычном общении с людьми, когда вы формулируете свой запрос примерно таким образом: “Найдите, пожалуйста, тех пациентов, которые не посещали нас на протяжении последних 6 месяцев”;

Работая с базой данных, вы вводите следующее:

```
SELECT last_name, first_name, last_visit FROM patient WHERE
```

```
last_visit < DATE_SUB(CURRENT_DATE, INTERVAL 6 MONTH);
```

Возможно, вы никогда не встречали ничего подобного, но перспектива получить ответ через секунду или две вместо изнурительного многочасового просмотра записей должна показаться привлекательной.

– гибкость поиска. Нет необходимости искать записи строго в соответствии с порядком, в котором они были записаны (по фамилии пациента, например). Информационной системе можно приказать расположить записи, отсортированные в любом порядке: по фамилии, названию страховой компании, дате последнего визита и т.д.;

– гибкость формата вывода. После того как необходимые записи найдены, копировать записи вручную не нужно. Можно сделать запрос информационной системе на вывод нужного списка. Иногда достаточно просто распечатать информацию. В других случаях вам может понадобиться воспользоваться этими данными в другой программе. (Например, после получения списка пациентов, которые забыли сделать ежегодный профилактический осмотр, эти данные можно переслать в текстовый редактор и уже на основании этой информации распечатать приглашения этим пациентам посетить ваш кабинет.) Предположим, что вам нужна только итоговая информация, такая как счетчик выбранных записей. И это совсем не обязательно делать вручную. Информационная система сгенерирует такой отчет сама;

– одновременный многопользовательский доступ к записям. Предположим, что сразу два человека хотят просмотреть одну запись. При бумажном способе ведения дел второй кандидат всегда вынужден ждать, пока первый закончит просмотр бумаг. СУБД позволяет получить доступ к одной и той же записи одновременно;

– удаленный доступ и передача записей в электронном виде. Бумажная технология ведения дел требует от вас быть там, где находятся сами бумаги, в противном случае кто-то должен скопировать и переслать их вам. Электронный способ ведения записей позволяет производить удаленный доступ к записям и передавать их в электронном виде. Предположим, что ваша стоматологическая фирма состоит из подразделений, удаленных территориально, электронный способ ведения учета позволяет получить доступ к вашим записям из удаленных офисов. Нет необходимости посылать копии записей о пациентах курьером, даже если кто-либо не имеет в своем распоряжении базы данных, аналогичной вашей, но имеет электронную почту. Ему можно будет послать содержимое базы данных в электронном виде[10];

Кроме того, в разработке используется фреймворк Kohana. Kohana – это PHP5 веб-фреймворк с открытым кодом, который использует архитектурную модель MVC(Hierarchical Model View Controller — Иерархические Модель-Контроллер-Вид). Его цели — быть безопасным, лёгким и простым в использовании.

Особенности:

– высокая скорость работы;

- безопасность;
- использование возможностей PHP5;
- большое количество встроенных инструментов;
- простота понимания;
- использование концепции HMVC;
- полная совместимость с UTF-8.

На основе модели VPwin можно построить модель данных. Для построения модели данных Logic Works предлагает мощный и удобный инструмент - ERwin. Хотя процесс преобразования модели VPwin в модель данных плохо формализуется и поэтому полностью не автоматизирован, Logic Works предлагает удобный инструмент для облегчения построения модели данных на основе функциональной модели - механизм двунаправленной связи VPwin – Erwin[12].

ERwin имеет два уровня представления модели - логический и физический. На логическом уровне данные представляются безотносительно конкретной СУБД, поэтому могут быть наглядно представлены даже для неспециалистов. Физический уровень данных - это, по существу, отображение системного каталога, который зависит от конкретной реализации СУБД. ERwin позволяет проводить процессы прямого и обратного проектирования БД. Это означает, что по модели данных можно сгенерировать схему БД или автоматически создать модель данных на основе информации системного каталога. Кроме того, ERwin позволяет выравнивать модель и содержимое системного каталога после редактирования того, либо другого. ERwin интегрируется с популярными средствами разработки клиентской части - PowerBuilder, SQLWindows, Visual Basic, Delphi, что позволяет автоматически генерировать код приложения, который готов к компиляции и выполнению[12].

CASE-технология является собой методологию разработки и проектирования автоматизированной системы (АС), а также комплекс инструментальных средств, которые дают возможность в наглядном виде моделировать предметную область, выполнять анализ этой модели на всех стадиях разработки и обслуживания АС и проектировать приложения в соответствии с информационными требованиями пользователей.

Множество ныне имеющихся CASE-средств базировано на методологиях структурного или объектно-ориентированного анализа и проектирования, которые используют спецификации в форме диаграмм или текстов для выражения внешних требований, взаимосвязей между моделями, тенденциям поведения поведения системы и структуры программных средств.

Визуальное моделирование в Rational Rose - процесс графического представления модели с помощью некоторого стандартного набора графических элементов. Наличие стандарта жизненно необходимо для реализации одного из преимуществ визуального моделирования — коммуникации. Общение между пользователями, разработчиками, аналитиками, тестировщиками, менеджерами и всеми остальными



участниками проекта является основной целью графического визуального моделирования.

Разработанные модели представляются всем интересующимся сторонам, которые хотят извлечь из них полезные данные. Глядя на модель, пользователи представляют свою работу с системой. Аналитики обнаружат взаимосвязь между элементами модели. Разработчики смогут понять, какие объекты можно создавать и что они должны делать. Тестировщики смогут отобразить взаимодействие между сущностями, а это даст им возможность построить тесты. Менеджеры посмотрят как на всю систему в целом, так и на взаимодействие ее отдельных компонентов. И наконец, руководители информационной отрасли, глядя на выстроенные модели, поймут, как взаимодействуют друг с другом системы в их организации. Поэтому, визуальные модели являются собой мощный инструмент, позволяющий продемонстрировать разрабатываемую систему всем заинтересованным лицам.

## **2 Проектирование и разработка АИС «Varpetit»**

### **2.1 Системный анализ объекта исследования**

Создание АИС «Varpetit» состоит из 4 стадий.

Стадия 1 – стадия технического задания. Сроки: 01.09.13 – 01.11.13.

Данная стадия состоит из следующих этапов:

- подготовительная работа – обследование и анализирование объекта, выбор модели для разработки;
- анализ требований к системе, анализ функций возможности системы, требование к интерфейсу;
- проектирование архитектуры системы.

Стадия 2 – стадия прототипирования (эскизного проекта). Сроки: 01.11.13 – 01.01.14.

Данная стадия состоит из следующих этапов:

- анализ, требование к программному обеспечению;
- проектирование архитектуры программного обеспечения;
- детальное проектирование программирования;
- выбор технологий.

Стадия 3 – стадия технический проект. Сроки: 01.01.14 – 01.04.14.

Данная стадия состоит из следующих этапов:

- кодирование и тестирование ПО;
- интеграция ПО (сборка всех компонентов);
- квалификационное тестирование системы.

Стадия 4 – стадия рабочий проект или же сдача проекта. Сроки 01.04.14 – 20.05.14.

Стадия состоит из этапов: установка и приемка ПО.

Основными целями создания АИС «Varpetit» являются:

- решение регламентных задач финансового учета заказов и материалов;
- снижение временных затрат на обработку заказов;
- получение отчетов о выполненных операциях, используя созданную базу данных;
- автоматическое ведение и обновление количественно-суммовой статистики;
- повышение эффективности и оперативности работы сотрудников кафе.

Для реализации поставленных целей система должна решать следующие задачи:

- увеличение контроля материальных потоков и ликвидация воровства;
- контроль финансовых потоков и прибыльности заведения;
- увеличение качества и скорости обслуживания посетителей;

– повышение лояльности за счет качества обслуживания, предоставления отличного уровня сервиса, поддержания отношений.

Задача увеличения контроля материальных потоков и ликвидации воровства решается как на уровне учетной (бэк-офисной), так и фронт-офисной системы. Все заказы посетителей попадают в систему, на основе них формируются заказы на приготовление блюд и выдача материалов в соответствии с принятой рецептурой блюда.

Задача контроля финансовых потоков и прибыльности заведения предназначена для формирования достоверной финансовой отчетности для менеджмента и собственника заведения.

Задача увеличения качества и скорости обслуживания посетителей предназначена для снижения среднего времени, необходимого для обработки заказа и повышение качества обслуживания с помощью электронного меню, скидочных программ.

Задача повышения лояльности предназначена для сохранения постоянных клиентов и привлечения новых гостей.

АИС «Vappetit» должна обеспечивать разграничение доступа к данным между администратором и обслуживающим персоналом кафе, для исключения воровства и мошенничества, а также обеспечивать безопасность данных, имеющих конфиденциальный характер.

## **2.2 Разработка UML-диаграмм АИС «Vappetit»**

### **2.2.1 Вид с точки зрения поведения**

На диаграмме прецедентов (вариантов использования) показано взаимодействие между вариантами использования и действующими лицами.

Она отражает требования к системе с точки зрения пользователя. Таким образом, варианты использования – это функции, выполняемые системой, а действующие лица – это заинтересованные по отношению к создаваемой системе[5].

Основная задача диаграммы вариантов использования - представлять собой единое средство, дающее возможность заказчику, конечному пользователю и разработчику совместно обсуждать функциональность и поведение системы.

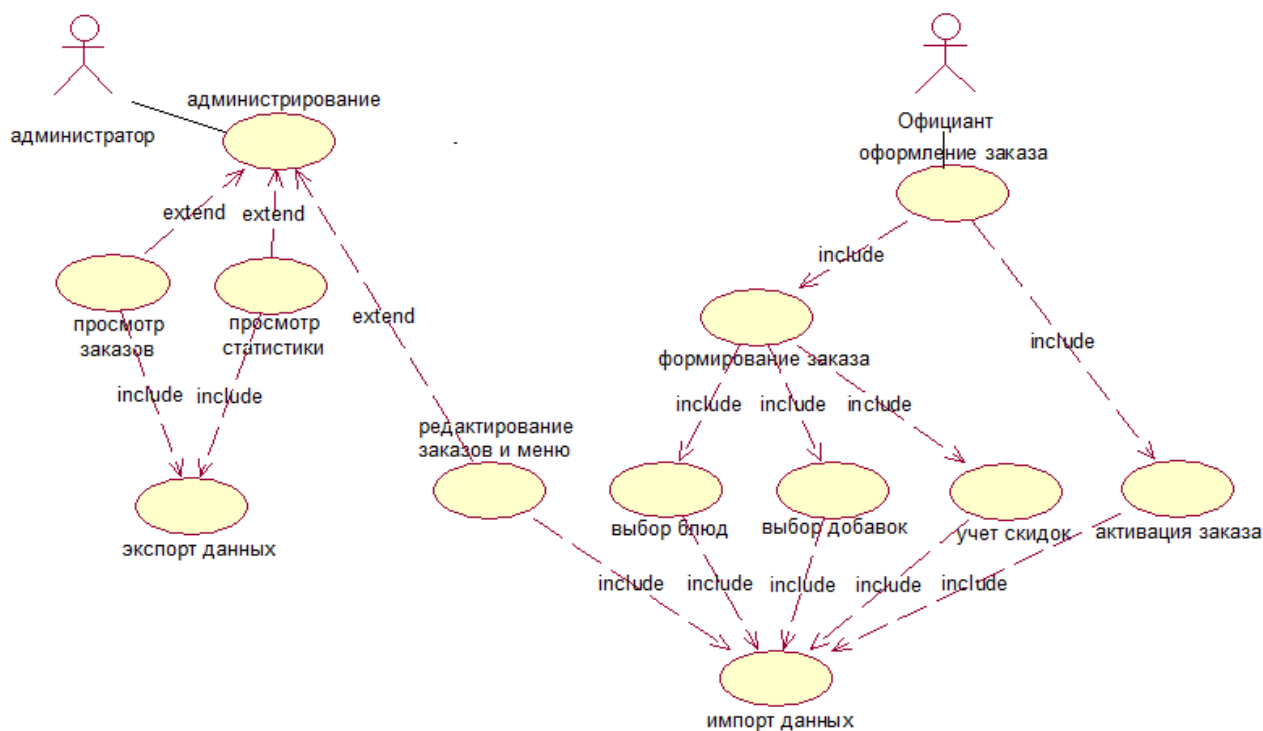


Рисунок 2.1 – Диаграмма прецедентов

На этой диаграмме два действующих лица. Существует также 4 основных действия, выполняемых моделируемой системой: администрирование, оформление заказа, экспорт данных, импорт данных.

Сценарий варианта использования «администрирование»

Вариант использования: администрирование.

Актеры: администратор.

Цель: редактирование и просмотр данных о заказах, статистики, меню.

Краткое описание: администратор имеет доступ к редактированию и просмотру данных о заказах, статистике, меню, определяет дисконтную политику, прослеживает тенденции продаж.

Тип: базовый.

Ссылки на другие варианты использования:

Расширяет себя до ВИ «просмотр заказов», «просмотр статистики», «редактирование заказов и меню».

Основной поток событий:

- администратор просматривает ленту заказов;
- администратор просматривает статистику по ассортименту;
- администратор редактирует данные по заказам и меню;

## 2.2.2 Вид с точки зрения процесса

Диаграммы видов деятельности - это один из пяти видов диаграмм, применяемых в UML для моделирования динамических аспектов поведения системы. Диаграмма видов деятельности - это, по существу, блок-схема,

которая показывает, как поток управления переходит от одной деятельности к другой[3].

Основными элементами диаграмм видов деятельности являются обозначения состояния («начало», «конец»), действия (овал) и момента синхронизации действий (линейка синхронизации, на которой сходятся или разветвляются несколько стрелок).

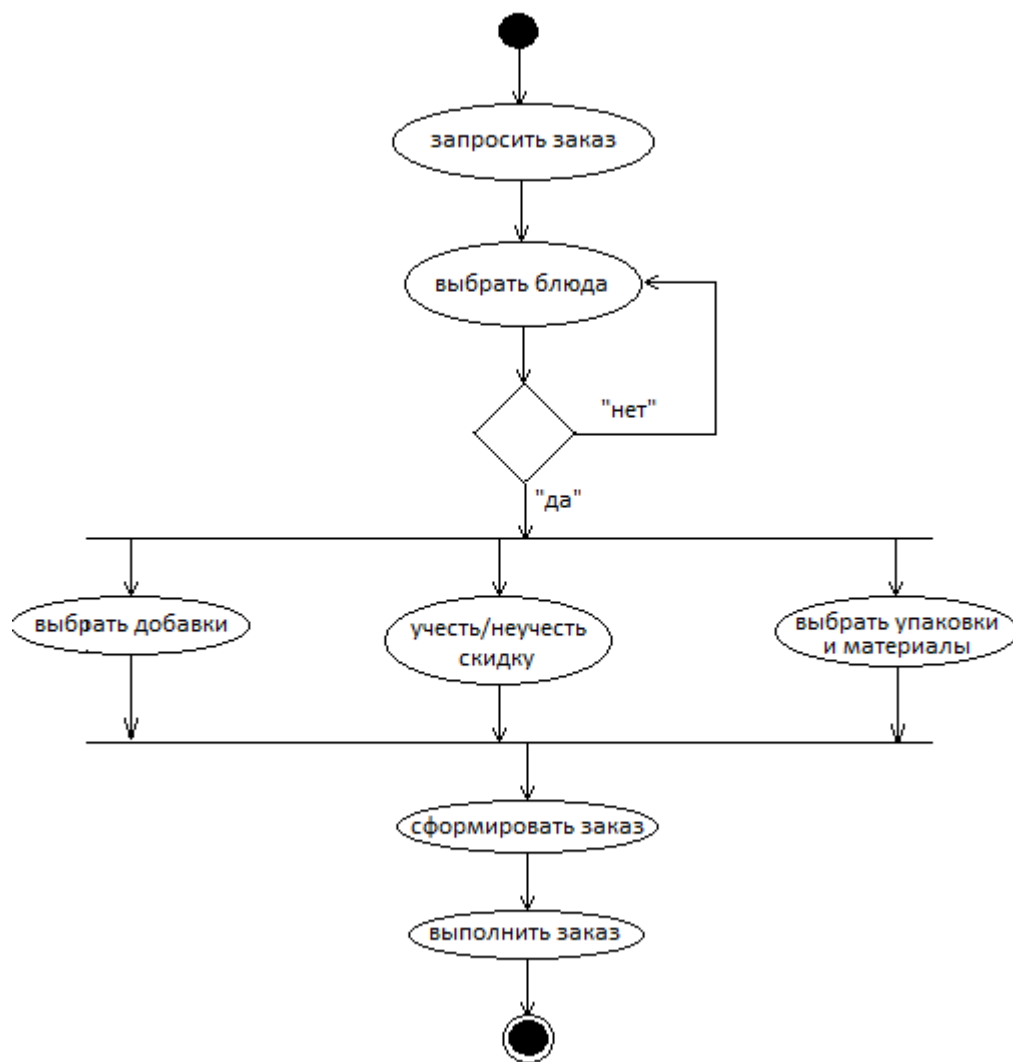


Рисунок 2.2 - Диаграмма видов деятельности для прецедента «Оформление заказа»

### 2.2.3 Вид с точки зрения проектирования

Диаграмма последовательности - удобное средство для обозначения очередности следования друг за другом различных стимулов (сообщений), с помощью которых объекты взаимодействуют между собой. Например, когда нужно проработать буквально по шагам какой-то очень важный участок выполнения программы[8].

Диаграмма деятельности обладает теми же общими свойствами, которые присущи всем остальным диаграммам: именем и графическим

наполнением, проецирующим ее на модель. От всех прочих диаграмму деятельности отличает ее специфичное содержание.

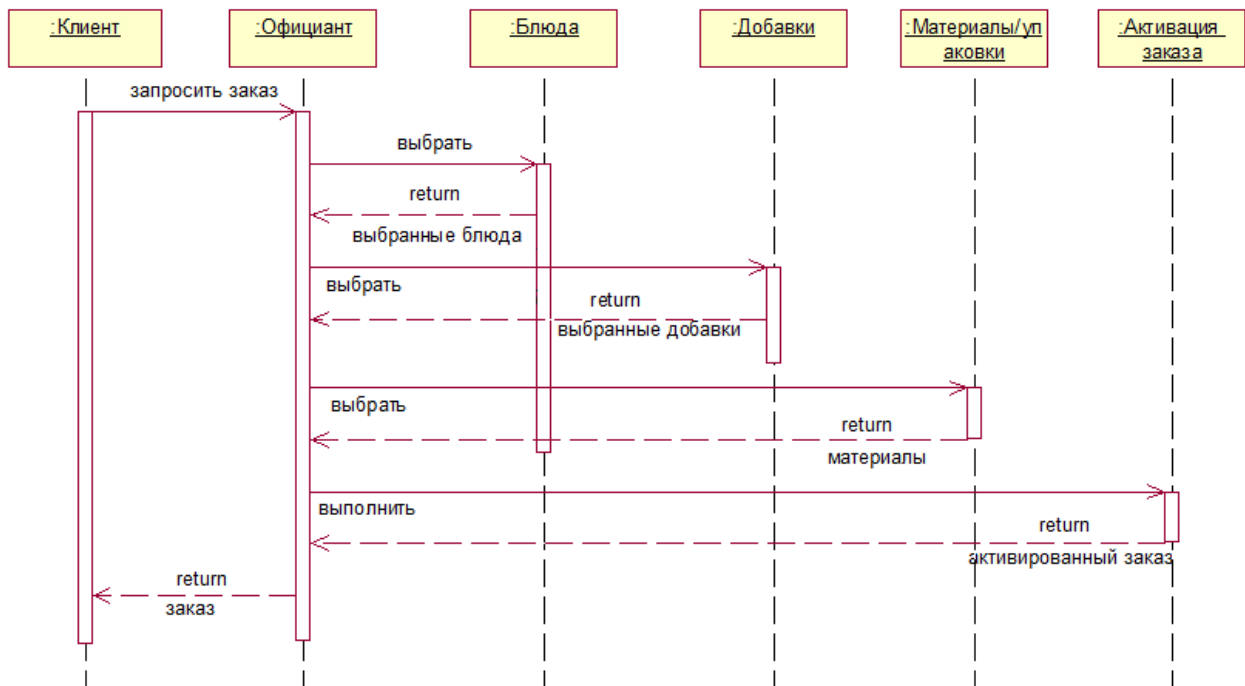


Рисунок 2.3 - Диаграмма последовательности прецедента «Оформление заказа»

Разберем каждый элемент диаграммы, по отдельности:

Объект, Участник (Object, Participant)

Обозначается прямоугольником, в котором указывается информация об участнике действий. Это, как правило, название объекта и его класс, разделенные двоеточием. Располагаются объекты (как правило) вдоль верхнего края диаграммы. От прямоугольника вниз спускается Линия Жизни.

Линия жизни (Life Line)

Линия, идущая вниз от участника, обозначающая отведенное объекту время жизни. Обозначается пунктирной линией.

Активация, фрагмент выполнения (Activation Bar, Execution Occurances)

Обозначается узким прямоугольником (серого или белого цвета), расположенным на линии жизни. Указывает начало и завершение действия, в котором участвует объект. Поскольку линия жизни - это метафора времени, то прямоугольник на линии жизни указывает на активизацию объекта во времени[5].

## 2.2.4 Вид с точки зрения реализации

Диаграммы состояний определяют все возможные состояния, в которых может находиться конкретный объект, а также процесс смены состояний объекта в результате наступления некоторых событий[5].

На диаграмме имеются два специальных состояния - начальное (start) и конечное (stop). Начальное состояние выделено черной точкой, оно соответствует состоянию объекта, когда он только что был создан. Конечное состояние обозначается черной точкой в белом кружке, оно соответствует состоянию объекта непосредственно перед его уничтожением. На диаграмме состояний может быть одно и только одно начальное состояние.

Основными элементами диаграммы состояний являются «Состояние» и «Переход». Диаграмма состояний имеет схожую семантику с диаграммой деятельности, только деятельность здесь заменена состоянием, переходы символизируют действия. Таким образом, если для диаграммы деятельности отличие между понятиями «Деятельность» и «Действие» заключается в возможности дальнейшей декомпозиции, то на диаграмме состояний деятельность символизирует состояние, в котором объект находится продолжительное количество времени, в то время как действие моментально[5].

Состояние может содержать только имя или имя и дополнительно список внутренних действий. Список внутренних действий содержит перечень действий или деятельностей, которые выполняются во время нахождения объекта в данном состоянии. Данный список фиксированный. Список основных действий включает следующие значения:

- entry - действие, которое выполняется в момент входа в данное состояние (входное действие);
- exit - действие, которое выполняется в момент выхода из данного состояния (выходное действие);
- do - выполняющаяся деятельность ("do activity") в течение всего времени, пока объект находится в данном состоянии
- defer - событие, обработка которого предписывается в другом состоянии, но после того, как все операции в текущем будут завершены.

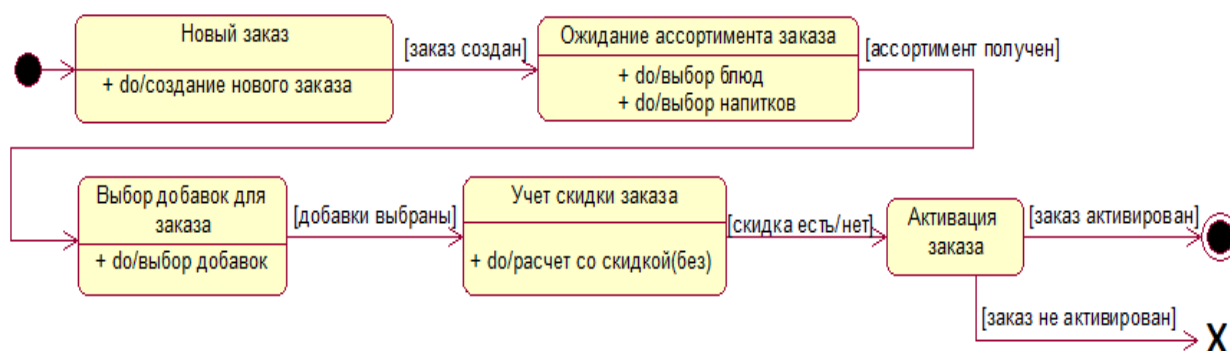


Рисунок 2.4 – Диаграмма состояний

Для представления физических сущностей в языке UML применяется специальный термин - компонент (component). Компонент реализует

некоторый набор интерфейсов и служит для общего обозначения элементов физического представления модели. Для графического представления компонента может использоваться специальный символ - прямоугольник со вставленными слева двумя более мелкими прямоугольниками. Внутри объемлющего прямоугольника записывается имя компонента и, возможно, некоторая дополнительная информация. Изображение этого символа может незначительно варьироваться в зависимости от характера ассоциируемой с компонентом информации.

В метамодели языка UML компонент является потомком классификатора. Он предоставляет организацию в рамках физического пакета ассоциированным с ним элементам модели. Как классификатор, компонент может иметь также свои собственные свойства, такие как атрибуты и операции.

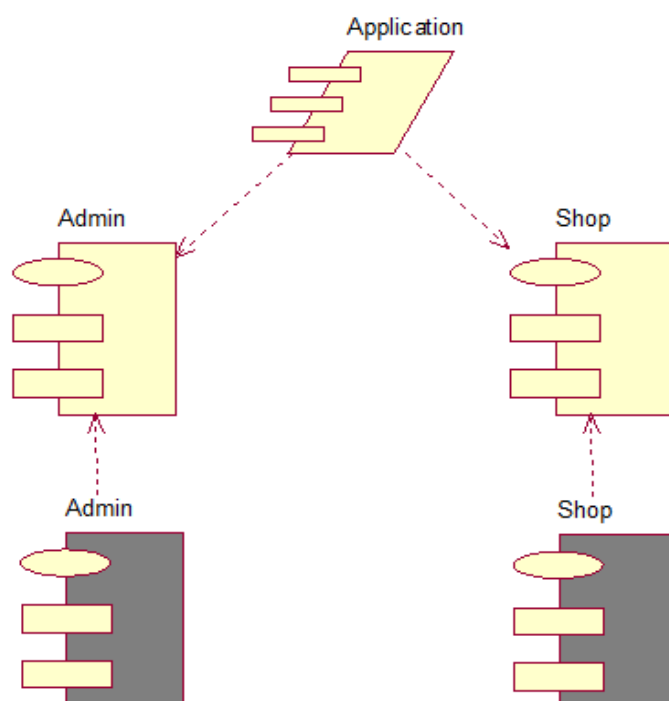


Рисунок 2.5 – Диаграмма компонентов

Главный компонент, который фактически управляет остальными это Application(приложение).

Каждый компонент состоит из двух частей:

- спецификация – это заголовочный файл для сведений о прототипах функций для класса (не закрашенная часть);
- тело пакета – часть, которая содержит код операции класса (закрашенная часть).

Диаграмма компонентов показывает, как выглядит модель на физическом уровне.



## 2.2.5 Вид с точки зрения развертывания

Физическое представление программной системы не может быть полным, если отсутствует информация о том, на какой платформе и на каких вычислительных средствах она реализована. Для представления общей конфигурации и топологии распределенной программной системы в UML предназначены диаграммы размещения[8].

По существу, диаграммы развертывания - это просто диаграммы классов, сосредоточенные на системных узлах.

Диаграмма размещения предназначена для визуализации элементов и компонентов программы, существующих лишь на этапе ее исполнения. При этом представляются только компоненты-экземпляры программы, являющиеся исполняемыми файлами или динамическими библиотеками. Те компоненты, которые не используются на этапе исполнения, на диаграмме развертывания не показываются. Так, компоненты с исходными текстами программ могут присутствовать только на диаграмме компонентов. На диаграмме размещения они не указываются[8].

Диаграмма размещения отражает физические взаимосвязи между программными и аппаратными компонентами системы. Она является хорошим средством для того, чтобы показать маршруты перемещения объектов и компонентов в распределенной системе. Каждый узел на диаграмме размещения представляет собой некоторый тип вычислительного устройства – в большинстве случаев, часть аппаратуры. Эта аппаратура может быть простым устройством или датчиком, а может быть и мэйнфреймом.

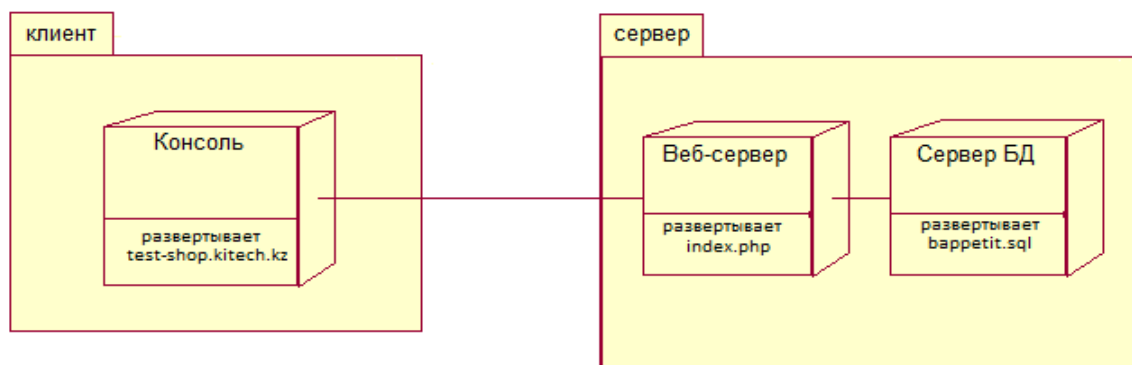


Рисунок 2.6 – Диаграмма развертывания

Диаграмма размещения содержит графические изображения отделов и связей между ними. В отличие от диаграмм логического представления, диаграмма размещения является единой для системы в целом, поскольку должна всецело отражать особенности ее реализации. Разработка диаграммы размещения, как правило, является последним этапом спецификации модели программной системы.

## 2.3 Разработка базы данных АИС «Vappetit»

Общим способом представления логической модели БД является построение ER-диаграмм (Entity-Relationship - сущность-связь). В этой модели сущность определяется как дискретный объект, для которого сохраняются элементы данных, а связь описывает отношение между двумя объектами.

Логический (концептуальный) уровень построен с учетом специфики и особенностей конкретной СУБД. Этот уровень представления данных ориентирован больше на компьютерную обработку и на программистов, которые занимаются ее разработкой. На этом уровне формируется концептуальная модель данных, то есть специальным способом структурированная модель предметной области, которая отвечает особенностям и ограничениям выбранной СУБД[9].

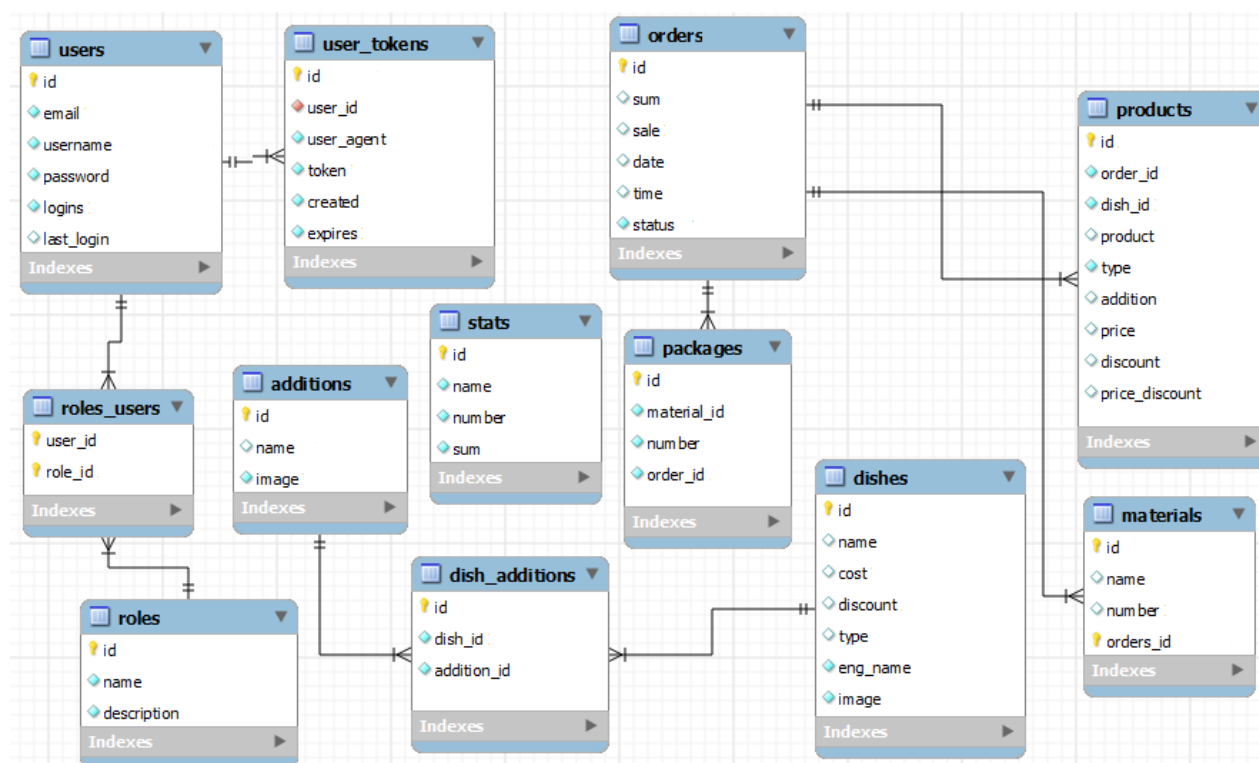


Рисунок 2.7 - Логический уровень

Физическая модель данных зависит от конкретной СУБД, фактически являясь отображением системного каталога. В физической модели содержится информация обо всех объектах БД. Поскольку стандартов на объекты БД не существует (например, нет стандарта на типы данных), физическая модель зависит от конкретной реализации СУБД. Следовательно, одной и той же логической модели могут соответствовать несколько разных физических моделей. Если в логической модели не имеет значения, какой конкретно тип данных имеет атрибут, то в физической модели важно описать всю информацию о конкретных физических объектах - таблицах, колонках,

индексах, процедурах и т.д. Разделение модели данных на логические и физические позволяет решить несколько важных задач[9].

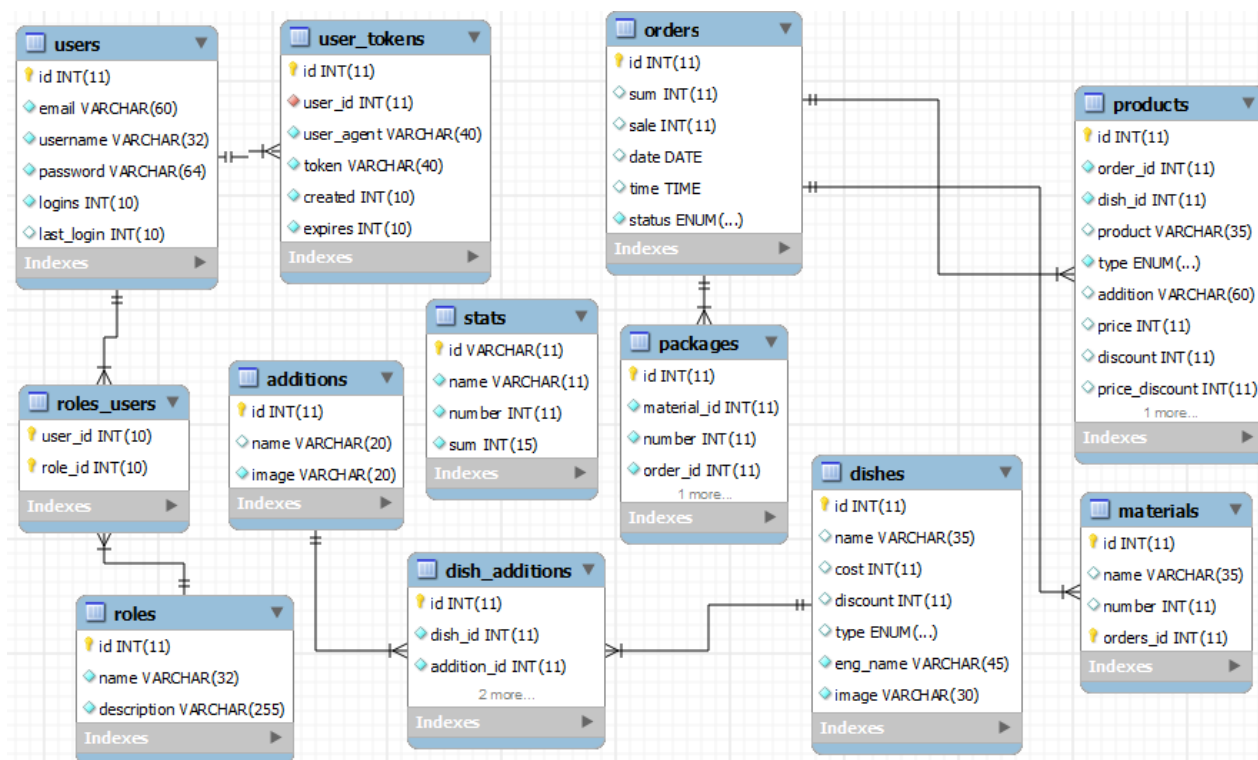


Рисунок 2.8 - Физический уровень

## **3 Разработка программного обеспечения АИС «Vappetit»**

### **3.1 Назначение и условия выполнения программы**

АИС «Vappetit» предназначена для автоматизации преимущественно бэк-офисной деятельности кафе «VonAppetit» в части исполнения следующих процессов:

- решение регламентных задач финансового учета заказов и материалов;
- снижение временных затрат на обработку заказов;
- получение отчетов о выполненных операциях, используя созданную базу данных;
- автоматическое ведение и обновление количественно-суммовой статистики;
- повышение эффективности и оперативности работы сотрудников кафе.

Основными целями создания АИС «Vappetit» являются:

- оперативность получения данных по товароучету и статистике;
- повышение эффективности исполнения процессов, перечисленных выше, путем сокращения непроизводительных и дублирующих операций, операций, выполняемых "вручную", оптимизации информационного взаимодействия участников процессов;
- повышение качества принятия управленческих решений за счет оперативности представления, полноты, достоверности и удобства форматов отображения информации;
- повышение информационной безопасности бизнес-данных и прозрачности деятельности рабочего персонала кафе «VonAppetit».

Входящие в состав АИС «Vappetit» задачи в процессе функционирования должны обмениваться информацией на основе открытых форматов обмена данными, используя для этого входящие в их состав модули информационного взаимодействия.

Для АИС «Vappetit» определены следующие режимы функционирования:

- нормальный режим функционирования;
- аварийный режим функционирования.

Основным режимом функционирования ИУС является нормальный режим.

В нормальном режиме функционирования системы:

- программное обеспечение и технические средства обеспечивают возможность круглосуточного функционирования, с перерывами на обслуживание;
- исправно работает оборудование;
- исправно функционирует программное обеспечение системы.

Для обеспечения нормального режима функционирования системы необходимо выполнять требования и выдерживать условия эксплуатации программного обеспечения и технических средств системы[16].

Аварийный режим функционирования системы характеризуется отказом одного или нескольких компонент программного и (или) технического обеспечения.

В случае перехода системы в аварийный режим необходимо:

- завершить работу всех приложений, с сохранением данных;
- выполнить резервное копирование БД.

После этого необходимо выполнить комплекс мероприятий по устранению причины перехода системы в аварийный режим.

АИС «Varpetit» должна реализовывать возможность дальнейшей модернизации как программного обеспечения.

Также необходимо предусмотреть возможность увеличения производительности системы путем её масштабирования.

Для эксплуатации АИС «Varpetit» определены следующие роли:

- администратор БД;
- пользователь.

Основными обязанностями администратора баз данных являются:

- установка, модернизация, настройка параметров программного обеспечения СУБД;
- оптимизация прикладных баз данных по времени отклика, скорости доступа к данным;
- разработка, управление и реализация эффективной политики доступа к информации, хранящейся в прикладных базах данных.

Администратор баз данных должен обладать высоким уровнем квалификации и практическим опытом выполнения работ по установке, настройке и администрированию используемых в ИУС СУБД[16].

Пользователи системы должны иметь опыт работы с персональным компьютером на базе операционных систем Microsoft Windows на уровне квалифицированного пользователя и свободно осуществлять базовые операции в стандартных Windows[16].

Рекомендуемая численность для эксплуатации АИС «Varpetit»:

- администратор - 1 штатная единица;
- пользователь - число штатных единиц определяется структурой предприятия.

АИС «Varpetit» должны обеспечивать возможность исторического хранения данных с глубиной не менее 10 лет.

Система должна обеспечивать:

- для операций навигации по экранным формам системы - не более 5 сек;
- для операций формирования отчетов и статистики - не более 10 сек.

Время формирования аналитических отчетов определяется их сложностью и может занимать продолжительное время.

Система должна предусматривать возможность масштабирования по производительности и объему обрабатываемой информации без модификации ее программного обеспечения путем модернизации используемых технических средств.

Система должна сохранять работоспособность и обеспечивать восстановление своих функций при возникновении следующих внештатных ситуаций:

- при сбоях в системе электроснабжения аппаратной части, приводящих к перезагрузке ОС, восстановление программы должно происходить после перезапуска ОС и запуска исполняемого файла системы;
- при ошибках в работе аппаратных средств (кроме носителей данных и программ) восстановление функции системы возлагается на ОС;
- при ошибках, связанных с программным обеспечением (ОС и драйверы устройств), восстановление работоспособности возлагается на ОС.

Для защиты аппаратуры от бросков напряжения и коммутационных помех должны применяться сетевые фильтры.

Все периферийные элементы технических компонентов системы, которая находится под напряжением, должны быть защищены от прикосновения, а сами технические средства должны быть снабжены занулением или защитным заземлением в соответствии с ГОСТа СТ РК 12.4.026-2002

Схема электропитания должна всегда обеспечивать безопасное отключение в случаях перегрузки и короткого замыкания в цепях нагрузки, а также аварийное отключение.

Разрабатываемая система должна отвечать требованиям безопасности, эргономики и профессиональной медицины при условии снабжения качественным оборудованием (ПЭВМ, монитор и пр.), имеющим соответствующие сертификаты соответствия и безопасности Казстандарта.

Физическая и техническая защита аппаратных узлов системы, носителей данных, безотказное и стабильное энергоснабжение, резервирование ресурсов, текущий сервис реализуется техническими и организационными способами, предусмотренными в ИТ инфраструктуре Заказчика.

В целях оптимальной эксплуатации системы должно быть обеспечено бесперебойное энергоснабжение ПК. При эксплуатации система должна соответствовать стандартам хранения носителей и использования ПК, влажность и температура воздуха.

Регулярное техническое обслуживание используемого ПК должно проводиться в соответствии с указаниями технической документации производителей, не менее одного раза в год.

Периодическое техническое тестирование и обслуживание ПК должны включать в себя обслуживание и тестирование ПК, кабельной системы, устройств бесперебойного питания.

В моменты проведения периодической технической проверки должен проводиться внешний и внутренний осмотр и чистка ПК, проверка контактов соединения, тестирование параметров настроек деятельности ПК.

Расположение ПК должно исключать возможность случайного и бесконтрольного проникновения в них других лиц и обеспечивать безопасность и сохранность находящихся в этих помещениях конфиденциальных документов и технических элементов.

Расположение ПК должно соответствовать требованиям техники безопасности, пожарной безопасности и санитарным нормам.

Пользователи системы должны соблюдать правила эксплуатации электронной вычислительной техники.

Квалификация сотрудников и его подготовка должны соответствовать техническим требованиям.

АИС должна обеспечивать защиту от несанкционированного доступа (НСД).

Элементы подсистемы защиты от несанкционированного доступа должны обеспечивать авторизацию и идентификацию пользователя.

Должны использоваться "слепые" пароли (при введении пароля его знаки не отображаются на экране либо преобразуются одним типом символов; количество символов не соответствует размеру пароля).

Программное обеспечение АИС «Vappetit» должно обеспечивать восстановление и функционирование при правильной перезагрузке аппаратных средств. Должна быть обеспечена возможность организации резервного дублирования данных СУБД.

Внедрение системы, как и установка отдельных компонентов системы не должна сопровождаться дополнительными требованиями к приобретению лицензий на ПО других изготовителей.

Экранные формы должны проектироваться с учетом требований унификации:

- все экранные формы пользовательского интерфейса должны быть выполнены в едином графическом дизайне, с одинаковым расположением основных элементов управления и навигации;
- для обозначения сходных операций должны использоваться сходные графические значки, кнопки и другие управляющие (навигационные) элементы.

Структура, состав и методы работы с данными в системе определяются на стадии технического проектирования.

Размещение информации должно реализовываться на базе современных реляционных или СУБД. Для организации целостности данных должны использоваться внутренние механизмы СУБД.

Доступ к данным должен быть предоставлен только авторизованным пользователям.

Элементы системы, обеспечивающие хранение данных, должны использовать современные технологические приемы, которые обеспечивают повышенную надежность хранения информации и своевременную замену оборудования.

Техническое обеспечение системы должно максимально и наиболее эффективным образом использовать существующий ПК у персонала кафе.

К эксплуатации системы могут допускаться сотрудники, умеющие работать на ПК, осведомленные с правилами эксплуатации и прошедшие курс обучения по работе с системой.

## **3.2 Разработка интерфейса АИС «Vappetit»**

### **3.2.1 Разработка логики работы программ и пользовательских интерфейсов**

Инструментарий для разработки пользовательского интерфейса, как правило, включает в себя библиотеку примитивов компонентов интерфейса (меню, кнопки, полосы прокрутки и др.) и предназначен для использования программистами.

Высокоуровневые средства разработки интерфейса могут быть использованы непрограммистами и снабжены языком, который позволяет специфицировать функции ввода-вывода, а также определять, используя технику непосредственного манипулирования, интерфейсные элементы.

Пользовательский интерфейс состоит из нескольких компонентов управления. Каждый компонент отображает следующие виды информации:

- текстовую (текстовые поля ввода, метки, подписи);
- числовую (поля числового ввода, различные индикаторы прогресса, ползунки);
- графическую (изображения, индикаторы);
- звуковую (звуковые сообщения, команды);
- бинарную (флажки, индикаторы, радиокнопки);
- и другие.

Средства управления могут реагировать на операции пользователя, реализованные с помощью таких средств, как: мышь, клавиатура, сенсорный экран, голоси др.

Одним из главных показателей элемента системы является его состояние. И в каждом состоянии элементы могут вести себя и реагировать отличным образом. Это объясняется необходимостью того, чтобы пользователь не привел систему в запрещенное состояние.

Структура интерфейса представлена в объектно-ориентированном виде. Объект – это сущность, у нее есть атрибуты и процедуры. Например, форма – авторизации. Атрибуты – поля, записи, информация. Процедуры(методы) –



кнопки или совершаемые действия.

Экранные формы должны проектироваться с учетом требований унификации:

- все экранные формы пользовательского интерфейса должны быть выполнены в едином графическом дизайне, с одинаковым расположением основных элементов управления и навигации;
- для обозначения сходных операций должны использоваться сходные графические значки, кнопки и другие управляющие (навигационные) элементы.

### 3.2.2 Проектирование пользовательского интерфейса

Данная система содержит следующие окна:

- окно входа;
- основная форма;
- окно администратора.

Окно входа и регистрации

В этом окне содержится поле для ввода почты и пароля. Логин всегда остается стандартным, так же как и пароль, эти данные были уже заложены при создании программы АИС «Varpetit»(рисунок 3.1).

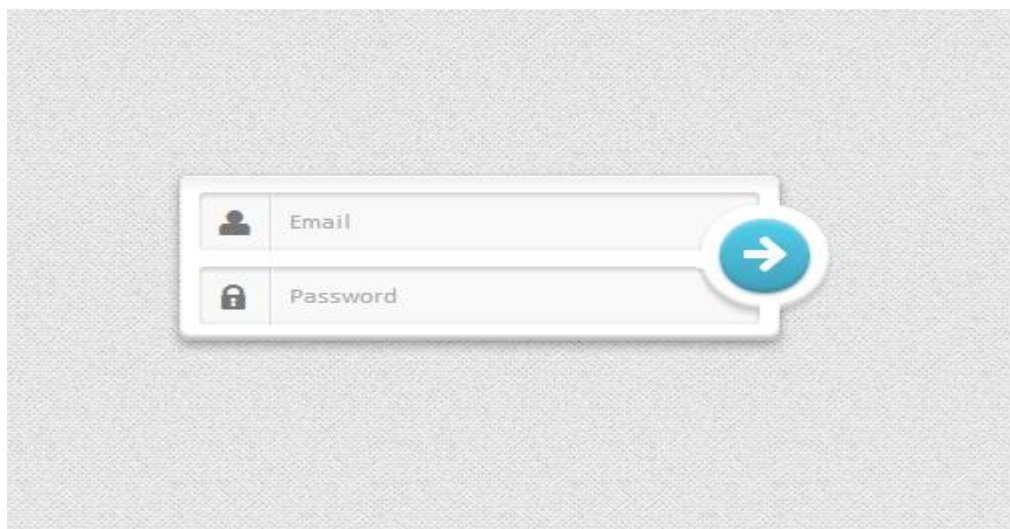


Рисунок 3.1 – Окно авторизации пользователя

Основная форма. В данном окне содержатся основные данные и функционал, необходимый для нормальной и корректной работы официанта(рисунок 3.2).



Рисунок 3.2 – Основная форма

Официант набирает выбранный ассортимент по заказу(рисунок 3.3).



Рисунок 3.3 – Выбор блюд

В результате в панели, расположенной справа отобразится комплекс заказанных блюд с их ценами. При необходимости, этот комплекс можно изменить, очистив список выбранных блюд или удалив только некоторые(рисунок 3.4).




Выход из приложения		
Заказ №1	12:54	03 May
	Бон-ланч	360 X
	Картофель фри	150 X
	Чай с молоком	60 X
	Кока-кола 0.5л	150 X
	Тосты	10 X
<b>Сумма заказа:</b>		<b>730</b>
<b>Текущие заказы:</b>		<b>0</b>
<b>Очистить</b>		<b>Продолжить</b>

Рисунок 3.4 – Панель выбранных блюд

После того, как для заказа выбраны необходимые блюда, официант переходит по кнопке «Продолжить» в окно «Корзина»(рисунок 3.5).

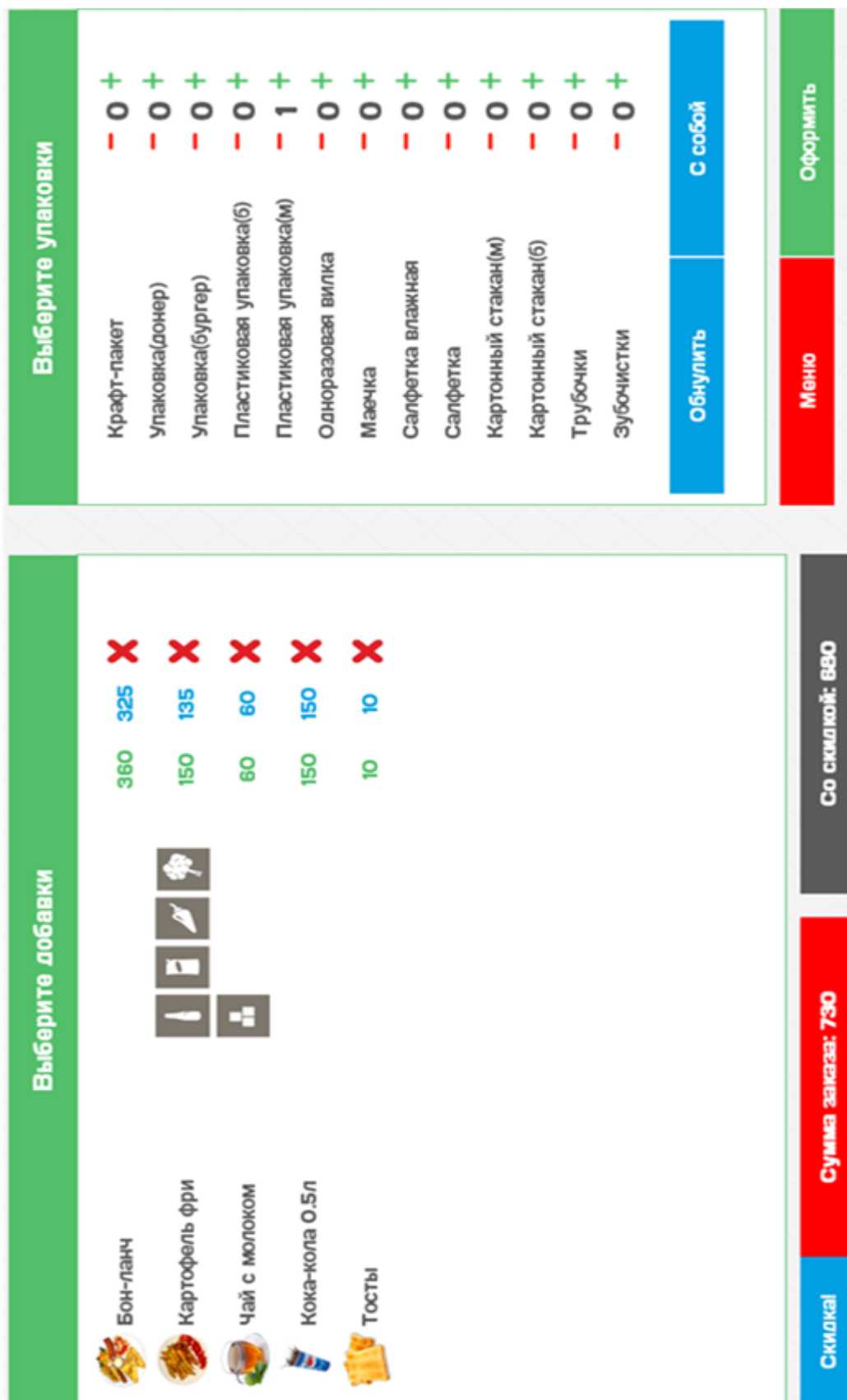












Рисунок 3.5 – Окно «Корзина»

В этом окне осуществляется выбор добавок к блюдам, выбор материалов для упаковки заказа и учет скидки(если она есть). Для некоторых блюд системой автоматически занесен предпочитаемый большинством клиентов набор добавок. Например, для чизбургера это кетчуп+майонез. Также, у блюд, для которых администратор предусмотрел скидку, она будет автоматически рассчитана. Как и в случае с добавками, для определенных позиций меню предусмотрен свой набор необходимых упаковок. Кроме того, если возникла необходимость снова изменить заказ, то из окна корзины повторно можно вернуться к меню и дополнительно выбрать или удалить блюда из заказа.

Выберите добавки

	Бон-ланч		360	325	✗
	Картофель фри	   	150	135	✗
	Чай с молоком		60	60	✗
	Кока-кола 0.5л		150	150	✗
	Тосты		10	10	✗

Скидка!

Сумма заказа: 730

Со скидкой: 680

Рисунок 3.6 – Панель выбора добавок и учета скидки

### Выберите упаковки

Крафт-пакет	- 0 +
Упаковка(донер)	- 0 +
Упаковка(бургер)	- 0 +
Пластиковая упаковка(б)	- 0 +
Пластиковая упаковка(м)	- 1 +
Одноразовая вилка	- 0 +
Маечка	- 0 +
Салфетка влажная	- 0 +
Салфетка	- 0 +
Картонный стакан(м)	- 0 +
Картонный стакан(б)	- 0 +
Трубочки	- 0 +
Зубочистки	- 0 +

Обнулить

С собой

Меню

Оформить

Рисунок 3.7 – Панель выбора материалов для упаковки заказа

После выполнения необходимых операций и нажатия кнопки «Оформить» заказ попадает в меню текущих заказов(рисунок 3.8).

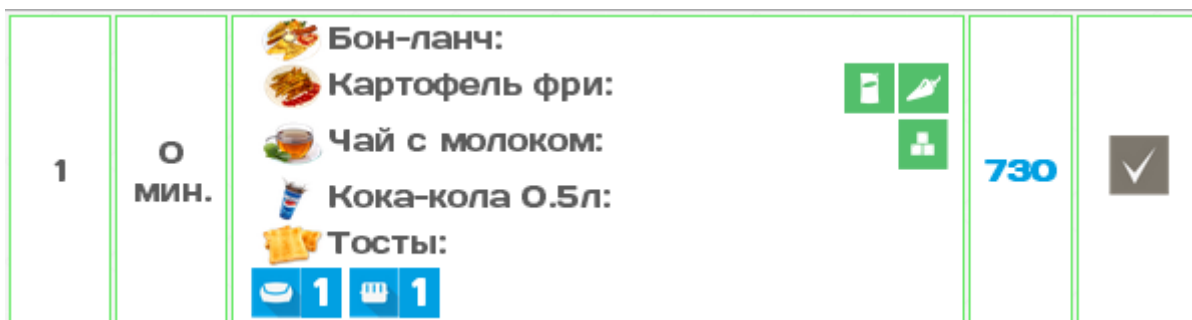


Рисунок 3.8 – Меню текущих заказов

В этом меню осуществляется активация текущих заказов(по нажатию галочки в правой части меню). Только после того, как заказ активирован, все необходимые данные поступают в базу данных. Данный пункт предусмотрен по причине того, что даже на этой стадии заказ могут потребовать изменить или вовсе отказаться от него. Поэтому данные о заказе заносятся в БД только после активации заказа (когда его преподнесли клиенту). В результате этого отсутствует вероятность нахождения в отчетах и статистике несуществующих или непринятых заказов.

Следует отметить, что после оформления заказа, система направляет вас в основное меню, а не в меню текущих заказов. Это сделано для возможности приема следующего заказа. А по мере доставки заказов клиентам, будут активироваться соответствующие позиции меню текущих заказов.

По окончании рабочего дня администратор может войти в окно администратора для просмотра необходимых данных по деятельности кафе. Доступ к окну администратора защищен паролем.

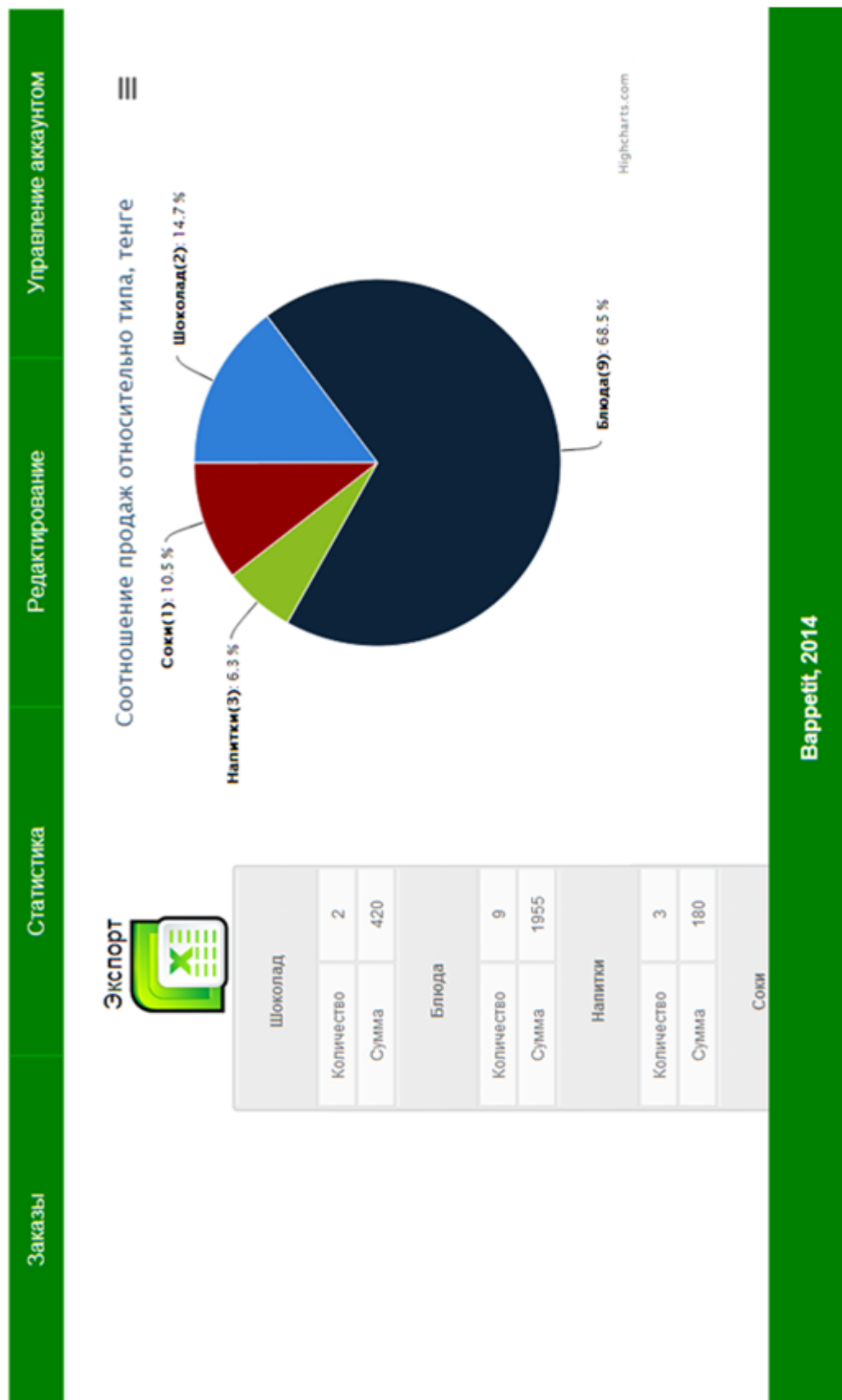













Рисунок 3.9 – Окно «Администратор»



На первой странице отображается суммарная статистика по основным позициям меню: блюда, напитки, соки, шоколад. Пройдя в «заказы» можно посмотреть отпущенные за день заказы, а также текущие(неактивированные) заказы. В «статистике» можно посмотреть, например, статистику блюд(рисунок 3.10).



Рисунок 3.10 – Окно «Администратор»(статистика блюд)

Название	Цена, тенге	Скидка, %	
 Американо	160	0	
 Американо с молоком	180	0	
 Американо с молоком(дабл)	300	0	
 Американо(дабл)	280	0	
			

Баррейт, 2014

Рисунок 3.11 – Окно «Администратор»(редактирование меню)

## **4 Экономическая часть**

### **4.1 Техничко-экономическое обоснование**

Программный продукт (АИС Варретит) данной дипломной работы представляет собой управленческую бэк-офисную часть решения для автоматизации кафе и заведений общественного питания. Данные решения позволяют вести товарный учет, рассчитывать калькуляцию и редактировать меню, получать различную аналитическую отчетность, ускорять взаимодействие во фронт-офисной части кафе и ресторанов.

Создание АИС Варретит преследует достижение следующих основных целей:

- решение регламентных задач финансового учета заказов и материалов;
- снижение временных затрат на обработку заказов;
- получение отчетов о выполненных операциях, используя созданную базу данных;
- автоматическое ведение и обновление количественно-суммовой статистики;
- повышение эффективности и оперативности работы сотрудников кафе.

Продукт автоматизирует финансовый учет в кафе и их сетях, предназначен для решения вопросов управления и гарантирует большую достоверность информации для принятия верных управленческих решений. Продукт направлен на своевременное реагирование на изменения различных факторов, воздействующих на деятельность ресторана (не обязательно финансовых). Анализ данных, дисконтная политика, планирование производства, анализ потребностей в ингредиентах, - это основные возможности продукта[1].

В любом кафе, будь то многоэтажный комплекс или кафе на 5 столиков, происходят одни и те же рабочие процессы. Автоматизация процессов призвана повысить прозрачность работы и, как следствие, прибыль заведения. В случае с небольшим кафе, контроль рабочих процессов позволяет сэкономить относительно большую часть прибыли, а организация автоматизации проще и дешевле.

Общая оценка рынка благоприятна для ПП подобного рода. Новые технологии расширяют функционал, сферы их применения и увеличивают число постоянных клиентов, сокращают злоупотребление среди рабочего персонала, облегчают работу заведующего производством[1].

Введение такой системы очень оправдано, в первую очередь, для небольших заведений, в которых недобросовестность персонала может очень чувствительно сказываться на доходе, который и так не слишком велик. И подобные заведения просто очень нуждаются в очень эффективных инструментах управления. Если правильно организовать дело с самого начала,

то даже небольшое заведение может очень быстро окупиться и начать давать хороший доход.

Пользователями системы являются любые сотрудники, имеющие в пользовании планшетное устройство.

Из выше сказанного можно сделать вывод, что разрабатываемый ПП значительно ускорит работу предприятий питания, чья деятельность связана с учетом большого числа различных факторов.

#### 4.2 Расчет трудоемкости разработки АИС «Vappetit»

Для определения трудоемкости разработки ПП приведен перечень всех основных этапов и видов работ, которые должны быть выполнены[6].

Форма разделения работ по этапам с указанием трудоемкости их выполнения приведена в таблице 4.1.

Таблица 4.1 - Распределение работ по этапам и видам и оценка их трудоемкости

ПП	Вид работы на данном этапе
	1. 2. для разработки 3. требований к системе
Дизайн проекта	Разработка эргономичного и интуитивного дизайна
Проектирование	1. 2. 3. архитектуры
Разработка	1. алгоритмов 2. элементов 3. отладка программы 4. документации к программному продукту
<b>ИТОГО трудоемкость выполнения проекта</b>	

Поскольку количество часов активной работы по разработке программного продукта равно 466, а в сутки на разработку выделялось девять

часов, следовательно, срок выполнения проекта равен 52 суткам. Для дальнейших расчетов время разработки программного продукта округляем до трех месяцев.

#### 4.3 Расчет затрат на разработку АИС «Vappetit»

Общая сумма затрат на материальные ресурсы ( $Z_M$ ) определяется по формуле:

$$Z_M = \sum P_i * C_i, \quad (4.1)$$

где  $P_i$  - расход  $i$ -го вида материального ресурса, натуральные единицы;  
 $C_i$  - цена за единицу  $i$ -го вида материального ресурса, тг;  
 $i$  - вид материального ресурса;

Расчет затрат на материальные ресурсы производится по форме, приведенной в таблице 4.2.

Наименование материального ресурса	Количество израсходованного материала	Цена за единицу, тг	Сумма, тг
1. умага писчая, пачка	1	1000	1000
2. артридж для принтера, шт.	1	5500	5500
3.	-	3500	3500

Таблица 4.2 - Затраты на материальные ресурсы

ругие канцтовары			
ИТОГО затраты на материальные ресурсы			10000

Общая сумма затрат на электроэнергию ( $Z_{э}$ ) рассчитывается по формуле:

$$Z_{э} = \sum M_i * K_i * T_i * Ц, \quad (4.2)$$

где  $M_i$  - паспортная мощность  $i$ -го электрооборудования, кВт;  
 $K_i$  - коэффициент использования мощности  $i$ -го электрооборудования (принят  $K_i=0.7$ );  
 $T_i$  - время работы  $i$ -го оборудования за весь период разработки, ч;  
 $Ц$  - цена электроэнергии, тг/кВт×ч;  
 $i$  - вид электрооборудования;

Затраты на электроэнергию приведены в таблице 4.3.

Таблица 4.3 - Затраты на электроэнергию

Наименование оборудования	Паспортная мощность, кВт	Коэфф. использования мощности	Время работы оборуд-я для разработки ПП, ч	Цена электроэнергии, $\frac{\text{тг.}}{\text{кВт} \times \text{ч}}$	Сумма, тг
1. К	0,4	0,7	466	15	1 958
ИТОГО затраты на электроэнергию					1 958

Общая сумма затрат на оплату труда ( $Z_{тр}$ ) определяется по формуле:

$$Z_{тр} = \sum ЧС_i * T_i, \quad (4.3)$$

где  $ЧС_i$  - часовая ставка  $i$ -го работника, тг;  
 $T_i$  - трудоемкость разработки ПП, чел.×ч;  
 $i$  - категория работника;

Часовая ставка программиста составляет 1 000 (тг/ч), трудоемкость разработки – 100 ч.

Часовая ставка старшего руководителя составляет 1 500 (тг/ч), трудоемкость разработки – 366 ч.

$$З_{ТР} = 1\,000 * 100 + 1\,500 * 366 = 649\,000 \text{ тг.}$$

Затраты на оплату труда приведены в таблице 4.4.

Таблица 4.4 - Затраты на оплату труда

Категория работника	Квалификация	Трудоемкость разработки ПП, ч	Часовая ставка, тг/ч	Сумма, тг
1 Программист	Программист 1 уровня	100	1 000	100 000
2 Старший руководитель	Менеджер по проектам	366	1 500	549 000
ИТОГО затраты на оплату труда				649000

Сумма годовых амортизационных отчислений определяется по формуле:

$$A = \text{Перв. стоимость} * \text{Норма амортизации}/100, \quad (4.4)$$

Амортизационные отчисления приведены в таблице 4.5.

Таблица 4.5 - Амортизация основных фондов (ОФ)

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Срок полезного использования оборудования и ПО, год	Сумма амортизации в год, тг	Сумма амортизации в месяц, тг
Монитор Samsung SyncMaster S19A100N	70 000	20	5	14 000	1167
Мышь оптическая	11000	20	5	2200	184

A4Tech BT-630-2 Bluetooth					
Клавиатура Kreolz KM01sb	8 000	20	5	1600	134
Принтер Canon Shot LBP-120	15 000	20	5	3000	250
Windows 7 SP 1 Rus	30 000	15	2,5	4 500	375
NetBeans 7.4	40 000	15	2,5	6 000	500
Denwer	Распространяется бесплатно				
MySQL	Распространяется бесплатно				
FileZilla	Распространяется бесплатно				
ИТОГО амортизация основных фондов					2610

Годовые нормы амортизации ОФ принимаются по налоговому кодексу РК или определяются, исходя из возможного срока полезного использования ОФ:

$$H_{Ai} = 100 / T_{Ni}, \quad (4.5)$$

где  $T_{Ni}$  - возможный срок использования  $i$ -го ОФ, год;

$$\begin{aligned}
 H_{A_{об}} &= 100 / 5 = 20; \\
 A_{мон} &= (70\,000 * 20) / 100 = 14\,000 \text{ тг}; \\
 A_{мышь} &= (11\,000 * 20) / 100 = 2\,200 \text{ тг}; \\
 A_{клав} &= (8\,000 * 20) / 100 = 1\,600 \text{ тг}; \\
 A_{принт} &= (15\,000 * 20) / 100 = 3\,000 \text{ тг}; \\
 A_{вин} &= (30\,000 * 15) / 100 = 4\,500 \text{ тг}; \\
 A_{netbeans} &= (40\,000 * 15) / 100 = 6\,000 \text{ тг}.
 \end{aligned}$$

Сумма амортизации за один месяц =  $A / 12$ .

Сумма амортизационных отчислений за три месяца равна 7830 тг.

В статью «Прочие затраты» включаются расходы на арендную плату, включая коммунальные платежи, затраты на лицензирование и сертификацию, расходы на рекламу, канцелярские и прочие хозяйственные расходы[6].

Стоимость аренды помещения на месяц равна 35000 тг. (в эту сумму включены коммунальные услуги).

Арендная плата рассчитывается по формуле:



$$AP = Ca * S, \quad (4.6)$$

где Ca – срок аренды;  
S – стоимость аренды за 1 месяц.

$$AP = 35\,000 * 3 = 105\,000 \text{ тг.}$$

Расходы на интернет, месячная оплата которого составляет 4500 тг равны:

$$P_{и} = 3 * 4\,500 = 13\,500 \text{ тг.}$$

Прочие хозяйственные расходы составляют 5 000 тг.

Прочие затраты = 105 000 + 13 500 + 5 000 = 123 500 тг.

Социальный налог, согласно Налоговому кодексу РК, составляет 11 % от затрат на оплату труда. Пенсионные отчисления не облагаются социальным налогом.

$$O_{с} = (Z_{тр} - O_{п}) * 0,11, \quad (4.7)$$

где  $O_{п}$  – отчисления в пенсионный фонд, 10% от ФОТ;  
 $Z_{тр}$  – затраты на оплату труда.

$$O_{с} = Z_{тр} * 10\% = 649\,000 * 0,1 = 64\,900 \text{ тг.}$$

$$O_{с} = (649\,000 - 64\,900) * 0,11 = 64\,251 \text{ тг.}$$

На основании полученных данных по отдельным статьям в таблице 4.6 приведена смета затрат на разработку ПП

Таблица 4.6 - Смета затрат на разработку ПП

Статьи затрат	Сумма, тг
1.	
Материальные	10 000
затраты, в том числе:	1 958
-	649 000
материалы	64 251
-	7830
электроэнергия	123 500

2. Затраты на оплату труда.	
3. Отчисле ния на социальн ые нужды.	
4. Амортиза ция основных фондов.	
5. Прочие затраты.	
ИТ	856
ОГО по смете	39

#### 4.4 Определение возможной (договорной) цены АИС «Vappetit»

Величина возможной (договорной) цены ПП должна устанавливаться с учетом эффективности, качества и сроков ее выполнения на уровне, отвечающем экономическим интересам заказчика (потребителя) и исполнителя[6].

Договорная цена ( $C_d$ ) для прикладных ПП рассчитывается по формуле:

$$C_d = Z_{\text{нир}} * (1 + (P/100)) , \quad (4.8)$$

где  $Z_{\text{нир}}$  - затраты на разработку ПП (из таблицы 4.6), тг;

$P$  - средний уровень рентабельности ПП. % (принято 25%).

$$C_d = 856\,539 * (1 + 0,25) = 1\,070\,674 \text{ тг.}$$

Цена реализации с учетом НДС рассчитывается по формуле:

$$C_p = C_d + C_d * \text{НДС} , \quad (4.9)$$

НДС, согласно Налоговому кодексу РК, составляет 12 %.

$$Ц_p = 1\,070\,674 + 1\,070\,674 * 0,12 = 1\,199\,155 \text{ тг.}$$

#### 4.5 Расчет срока окупаемости АИС «Vappetit»

До внедрения АИС «Vappetit» в кафе работали 5 человек в смену в односменном режиме (3 официанта, повар, продавец-кассир) с месячным фондом оплаты труда 280000 тг.

После внедрения эти же функции могут выполнять 2 официанта, а с продавца-кассира снимается часть обязанностей. В этом случае расходы на оплату труда составят 200000 тг в месяц, и для предприятия это будет обходиться на 87920 тг (с учетом соц. отчислений) меньше ежемесячно, чем до внедрения системы.

Определим дисконтированный срок окупаемости ПП:

$$DPP = \sum (CF_t / (1+r)^t) \geq I_0, \quad (4.10)$$

где  $CF_t$  – приток денежных средств в период  $t$ ;

$r$  – коэффициент дисконтирования;

$I_0$  – величина исходных инвестиций в нулевой период.

Коэффициент дисконтирования  $r=20\%$ .

Доходы от внедрения:

В первом году:  $PV1 = 12 * 87920 / (1+0.2)^1 = 879200$  тг.

Во втором году:  $PV2 = 12 * 87920 / (1+0.2)^2 = 732666$  тг.

Сумма  $PV1$  и  $PV2$ :  $879200 + 732666 = 1611866$  тг.

Остаток =  $(1 - (1611866 - 1199155) / 732666) = 0,56$  года. Срок окупаемости инвестиций равен 1,56 года.

#### 4.6 Оценка социально - экономических результатов функционирования АИС «Vappetit»

Каждый день в кафе — это пересечение множества процессов. Простой пример: блюдо, допустим, блины, заказывают посетителем. Затем его берет повар и несет на кухню. Затем это блюдо попадает к клиенту, который, если не понравится, может еще и от него отказаться. А теперь представим, что таких блюд множество. И их сочетаний множество. Проследить за использованием нескольких блюд хороший менеджер способен, а вот большой масштаб одному человеку не под силу. Даже если хозяин доверяет персоналу (что в нашей действительности случается редко), и нет проблем с воровством, ПП позволит как минимум вовремя определить тенденцию тех же

заказов. С помощью ПП можно легко и оперативно определить, какие позиции меню прибыльные, а какие — не очень. Владелец заведения получает возможность быстро получать правдивую финансовую информацию о своем детище, а, значит, упрощается принятие бизнес-решений.

В результате проделанной работы затраты на разработку ПП составили 856 539 тг., а цена реализации(с учетом НДС) 1 199 155 тг. Поскольку срок окупаемости больше года, то окупаемость ПП вычислена с учетом дисконтирования. Срок окупаемости составил 1,56 года.

## **5 Безопасность жизнедеятельности**

### **5.1 Анализ условий труда в рабочем помещении**

#### **5.1.1 Анализ помещения**

Важную роль играет правильная организация условий труда в рабочем помещении. Таким образом, необходимо создать благоприятный микроклимат, полноценное освещение и учесть эргономические требования по организации рабочего места персонала.

В помещении на официанта и посетителей могут негативно действовать следующие физические факторы:

- повышенная и пониженная температура воздуха;
- чрезмерная запыленность и загазованность воздуха;
- повышенная и пониженная влажность воздуха;
- недостаточная освещенность рабочего места;
- превышающий допустимые нормы шум.

Биологические вредные производственные факторы в данном помещении отсутствуют.

К психологически вредным факторам, воздействующим на официанта в течение его рабочей смены можно отнести следующие:

- нервно-эмоциональные перегрузки;
- умственное напряжение;
- перенапряжение зрительного анализатора.

Помещение имеет следующие параметры:

– в рассматриваемом помещении работают два человека и обслуживается до 15 посетителей. Внешних источников шума, влияющих на процесс работы – нет;

– находится на 1 этаже пятиэтажного здания, размеры помещения (комнаты): длина 8 м, ширина 4 м, высота 3 м;

– два окна размером 1,5\*1,5;

– вид светопропускающего материала – стекло листовое, двойное;

– вид переплета – металлический;

– солнцезащитные устройства – убирание регулируемые жалюзи;

– в помещении допустимые параметры микроклимата;

– энергозатраты 121—150 ккал/ч (140—174 Вт);

– искусственное освещение – лампы люминесцентные серии ЛБ 80-4.

#### **5.1.2 Характеристика оборудования**

**Планшет Samsung Galaxy Tab 2 7.0 P3100:**

- операционная система: Android;
- поддержка: Android 4.2;
- процессор/чипсет: TI OMAP4430 1000 МГц;
- количество ядер: 2;
- оперативная память: 1Гб;
- встроенная память: 8Гб;
- поддержка карт памяти: microSDHC, до 32 Гб;
- экран: 1024x600;
- тип экрана: TFT PLS, глянцевый;
- сенсорный экран: емкостный, мультитач;
- число пикселей на дюйм: 170;
- видеопроцессор: PowerVR SGX540;
- емкость аккумулятора: 4000 мАч;
- размеры: 194x122x11мм;
- вес: 344 г;
- материал корпуса: пластик.

**Осветительное оборудование:**

- лампы люминесцентные серии ЛБ 80-4;
- электропитание: переменное напряжение 220-250В, частотой 50 Гц.

Электротехническое оборудование является потенциальным источником возникновения пожарной опасности. Оборудование малозумящее – вредность в качестве повышенного шума отсутствует. Все оборудование расположено на столе, либо на стенах помещения, что не создает помех в работе сотрудников.

### 5.1.3 Схема помещения

Схема помещения приведена на рисунке 5.1.

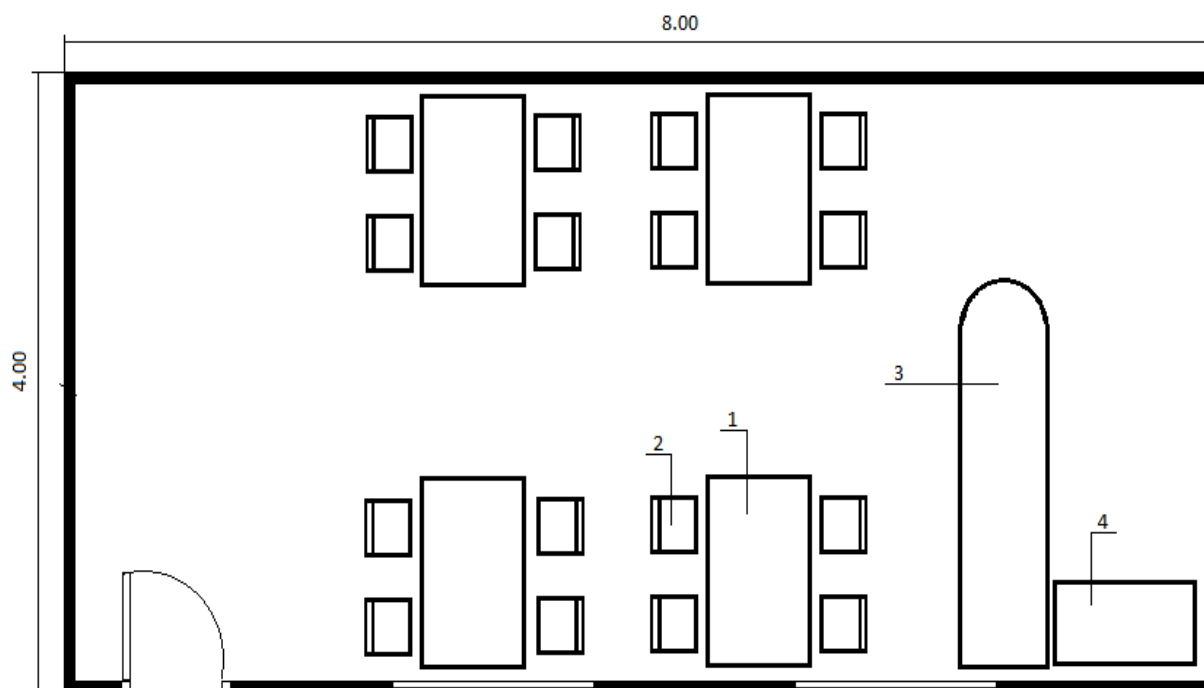


Рисунок 5.1 - План помещения. 1 – стол, 2 – стул, 3 – барная стойка, 4 – стол

### 5.1.4 Анализ системы искусственного освещения

Правильно спроектированное и выполненное производственное освещение улучшает условия зрительной работы, снижает утомляемость, способствует повышению производительности труда, благотворно влияет на производственную среду, оказывая положительное психологическое воздействие на работающего, повышает безопасность труда и снижает травматизм[7].

Требования к освещенности в помещениях, где установлены компьютеры, следующие: при выполнении зрительных работ средней точности общая освещенность должна составлять 200 лк, а комбинированная - 400 лк. Всеобщим межотраслевым документом, содержащим нормы естественного и искусственного освещения предприятий, является СНиП 23-05-95.

Кроме того все поле зрения должно быть освещено достаточно равномерно – это основное гигиеническое требование. Иными словами, степень освещения помещения и яркость экрана компьютера должны быть примерно одинаковыми, т.к. яркий свет в районе периферийного зрения значительно увеличивает напряженность глаз и, как следствие, приводит к их быстрой утомляемости[7].

#### 5.1.5 Параметры вентиляции

Грамотное проектирование систем вентиляции в кафе и ресторанах играет не менее важную роль, чем, например, подбор технологического оборудования для кухни. На предприятиях общественного питания вентиляция жизненно необходима, ведь она обеспечивает оптимальные санитарно-гигиенические параметры воздуха не только для персонала кухни, но и создаёт комфортные условия для посетителей.

Искусственная вентиляция создаётся при помощи электромеханических устройств - вентиляторов. Это искусственная или механическая вентиляция. На предприятия общественного питания невозможно обойтись лишь созданием оптимальных условий для естественной вентиляции. Поскольку в механической системе используется вентилятор, фильтр, воздухонагреватель и другие элементы, позволяющие перемещать, очищать и нагревать воздух, такие системы могут поддерживать комфортные условия в помещениях независимо от времени года и условий среды.

### 5.2 Расчет системы кондиционирования

#### 5.2.1 Расчет систем кондиционирования и вентиляции

Составление теплового и влажностного баланса для кондиционируемого помещения производится общеизвестными методами, принятыми в отопительно-вентиляционной технике. Здесь должны быть учтены все факторы, влияющие на изменение состояния воздушной среды помещения.

В комнатах различного назначения происходят в основном тепловые нагрузки, которые возникают снаружи помещения; тепловые нагрузки, возникающие внутри помещения(внутренние).

Наружные тепловые нагрузки представлены следующими составляющими:

- теплопотери или теплопоступления в результате инверсии температур через стены, потолки, полы, окна и т.д;
- разность температур снаружи помещения и внутри летом является плюсовой, в результате чего присутствует приток тепла снаружи во внутрь здания; и наоборот – зимой эта разность является отрицательной и направление потока изменяется;
- тепло от солнечного излучения через застекленные поверхности; данное явление проявляется в виде ощущаемого тепла;



– теплоступления от инфильтрации.

Расчет поступления тепла через внешние ограждающие конструкции в летний период года затрудняется существенными колебаниями температуры наружного воздуха в течение суток и еще большими колебаниями теплового потока на наружных поверхностях ограждений за счет солнечного излучения. Значительное влияние на теплообмен оказывает и массивность ограждений, благодаря чему колебания температуры на их внутренней поверхности уменьшаются.

Потери тепла через ограждающие конструкции в зимний период года рассчитывают в предположении стационарного режима, так как зимой значительных колебаний температуры наружного воздуха и особенно колебаний температуры на наружной стороне ограждений не наблюдается.

В зависимости от времени года и времени суток наружные тепловые нагрузки могут быть положительными.

Теплоступления и теплотери в результате разности температур определяются по формуле (5.1)

$$Q_{огр} = V_{пом} \cdot X_o \cdot (t_{Нрасч} - t_{Врасч}), \quad (5.1)$$

где  $V_{пом}$  – объем помещения,  $m^3$ :

$X_o$  – удельная тепловая характеристика,  $Вт/м^3 \cdot ^\circ C$ :

$t_{Нрасч}$  – наружная температура;

$t_{Врасч}$  – внутренняя температура, выбирается с учетом комфортных условий или технологических требований, предъявляемых к производственным процессам.

$$V_{пом} = 8 \cdot 4 \cdot 3 = 96 \text{ м}^3;$$

$$X_o = 0.42 \text{ Вт} / \text{м}^3 \cdot ^\circ C.$$

Для холодного периода – средняя температура самого холодного месяца в 13 часов:

$$t_{Нрасч} = -10 \text{ } ^\circ C.$$

Для холодного периода:

$$t_{Врасч} = 20 \text{ } ^\circ C.$$

$$Q_{огр} = 96 \cdot 0,42 \cdot (-10-20) = 1209,6 \text{ Вт};$$

Для теплого периода:  $Q_{огр} = 0 \text{ Вт}$

Избыточная солнечная теплота в зависимости от вида стекла почти до 90% поглощается средой здания, оставшаяся часть отражается. Самая большая тепловая нагрузка возникает при максимальном излучении, которое

делится на прямую и рассеянную составляющие. Сила излучения зависит от площади локализации, времени года и времени суток.

Теплопоступление от солнечного излучения через остекление определяется по формуле (5.2)

$$Q_p = (q^I F_o^I + q^{II} F_o^{II}) \cdot \beta_{c.z.}, \quad (5.2)$$

где  $q^I, q^{II}$  – тепловые потоки от прямой и рассеянной солнечной радиации, Вт/м<sup>2</sup>;

$F_o^I, F_o^{II}$  – площади светового проема, облучаемые и необлучаемые прямой солнечной радиацией, м<sup>2</sup>;

$\beta_{c.z.}$  – коэффициент теплопропускания.

$\beta_{c.z.} = 0,15$

Если нет наружных затеняющих козырьков и т. д. для периода облучения остекления солнцем, когда его лучи проникают через окно в помещение  $F_o^I = F_o; F_o^{II} = 0$ :

$$Q_p = q^I F_o \cdot \beta_{c.z.} = (q_{en} + q_{ep}) \cdot K_1^c \cdot K_2 \cdot \beta_{c.z.} \cdot n \cdot S_o, \quad (5.3)$$

где  $q_{en}, q_{ep}$  – тепловые потоки от прямой и рассеянной радиации, Вт/м<sup>2</sup>.

$F_o = nH_0B_0 = 2 \cdot 1,5 \cdot 1,5 = 4,5$  м<sup>2</sup> – площадь светового проема ( $n$  – число окон, высота  $H_0$  и ширина  $B_0$ );

$K_1$  – коэффициент затемнения остекления переплетами ( $K_1^c$  – для облученных проемов);

$K_2$  – коэффициент загрязнения остекления.

По таблице для широты в 44° СШ после полудня в 13-14 ч. при расположении С:

$q_{en} = 0$  Вт/м<sup>2</sup>;  $q_{ep} = 60$  Вт/м<sup>2</sup>;

$K_1^c = 0,72$ ;

$K_2 = 0,95$ .

Тогда, для С:

$$Q_p = (0 + 60) \cdot 0,72 \cdot 0,95 \cdot 0,15 \cdot 4,5 = 27,702 \text{ Вт}$$

## 5.2.2 Внутренние тепловые нагрузки

Внутренние нагрузки складываются из тепла:

- излучаемого людьми;
- излучаемого лампами и осветительными приборами;
- излучаемого компьютерами, печатающими устройствами и пр.

Теплопоступления от людей зависят от выполняемой работы и характеристик окружающего воздуха. Тепло, излучаемое человеком, состоит из ощутимого, то есть посылаемого в воздух помещения посредством конвекции и лучеиспусканий, и скрытого тепла, идущего на испарение влаги с кожи и из легких[7].

Общее количество тепла при низких температурах окружающей среды выше, чем при высоких; при низких температурах окружающей среды значение явного тепла значительно выше показателей скрытого тепла, при высоких температурах преобладает выделение скрытого тепла; с повышением физических нагрузок начинает преобладать выделение скрытого тепла.

Для теплого периода:

При  $t=26$  °С в помещении двое женщин при положении стоя либо легком движении выделяют явного тепла  $108 \text{ Вт}$ , а общего –  $226 \text{ Вт}$ . В среднем 7 посетителей(4 женщины и 3 мужчин) в положении сидя выделяют явного тепла  $405 \text{ Вт}$ , а общего –  $848 \text{ Вт}$ . Итого:

$$Q_{\text{л}}^{\text{я}} = 513 \text{ Вт}.$$

А выделение общего тепла:

$$Q_{\text{л}}^{\text{о}} = 1074 \text{ Вт}.$$

Для холодного периода:

При  $t=22$  °С в помещении двое женщин при положении стоя либо легком движении выделяют явного тепла  $143 \text{ Вт}$ , а общего –  $225 \text{ Вт}$ . В среднем 7 посетителей(4 женщины и 3 мужчин) в положении сидя выделяют явного тепла  $538 \text{ Вт}$ , а общего –  $845 \text{ Вт}$ . Итого:

$$Q_{\text{л}}^{\text{я}} = 681 \text{ Вт}.$$

А выделение общего тепла:

$$Q_{\text{л}}^{\text{о}} = 1070 \text{ Вт}.$$

Теплопоступление от осветительных приборов, оргтехники и оборудования рассчитывается следующим образом. Теплопоступление от ламп определяется по формуле (5.4)

$$Q_{\text{осв}} = \eta \cdot N_{\text{осв}} \cdot F_{\text{пол}}, \quad (5.4)$$

где  $\eta$  – коэффициент перехода электрической энергии в тепловую (для люминисцентных ламп  $\eta=0,5-0,6$ );

$N_{\text{осв}}$  – установленная мощность ламп ( $N=56 \text{ Вт/м}^2$ );

$F_{\text{пол}}$  – площадь пола:

$$F_{\text{пол}} = 8 \cdot 4 = 32 \text{ м}^2;$$

Тогда:

$$Q_{\text{осв}} = 0,5 \cdot 56 \cdot 32 = 896 \text{ Вт}.$$

Тепло, выделяемое производственным оборудованием, определяется по формуле (5.5)

$$Q_{об} = N_{уст} \cdot K, \quad (5.5)$$

$$Q_{об} = 0,95 \cdot 0,09 \cdot 1 = 0,0855 \text{ кВт};$$

Теплопритоки, возникающие за счет находящейся оргтехники, - это 30% мощности оборудования:

$$Q_{орг} = 0,0855 \cdot 0,3 = 0,02565 \text{ кВт}.$$

### 5.2.3 Расчет теплового баланса помещения

На основании выполненных расчетов составим баланс тепlopоступлений в помещении:

$$Q_{изб} = Q_{орг} + Q_p + Q_l^a + Q_{осв} + Q_{об} - Q_{опр}, \quad (5.6)$$

Для холодного периода:

$$Q_{избХ} = 25,65 + 27.702 + 681 + 896 + 85,5 - 1209,6 = 506,252 \text{ Вт};$$

Для теплого периода:

$$Q_{избТ} = 25,65 + 27.702 + 513 + 896 + 85,5 = 1715,852 \text{ Вт}.$$

Т.к.  $Q_{избХ} < Q_{избТ}$ , дальнейший расчет ведем по  $Q_{избТ}$ .

Количество воздуха, необходимого для подачи в помещении, исходя из теплового баланса, определяется по формуле (5.7)

$$G = \frac{Q_{изб}}{c \cdot (t_n - t_{пр})}, \quad (5.7)$$

где  $Q_{изб}$  – количество явного тепла (избыточного), передаваемого в помещение различными источниками, кДж/ч;

$c$  – весовая теплоемкость воздуха, 1,05 кДж/кг·°С;

$t_n$ ,  $t_{пр}$  – температура воздуха соответственно в помещении и на притоке.

Рассчитаем значение теплонапряженности воздуха  $Q_H$ :

$$Q_H = Q_{избТ} \cdot 860 / V = 1,715 \cdot 860 / 96 = 15.36 \text{ Ккал} / \text{м}^3;$$

Т.к  $Q_n < 20 \text{ Ккал/м}^3 \Rightarrow \Delta t = t_n - t_{np} = 6^\circ\text{C}$ .

Таким образом:

$$G = 15,36 / (1,05 \cdot 6) = 2,44 \text{ кг/ч}.$$

Определение расхода приточного воздуха, необходимого для поступления в помещение определяется по формуле (5.8)

$$L = \frac{(Q_{изб} \cdot 860)}{c \cdot \Delta t \cdot \gamma}, \text{ м}^3/\text{ч}, \quad (5.8)$$

где  $C$  – теплоемкость воздуха,  $0,24 \text{ ккал/кг}^0\text{C}$ ;

$\Delta t$  - выбираем в зависимости от теплонапряженности воздуха  $Q_n$  ;

$\gamma$  - удельная масса приточного воздуха,  $1,206 \text{ кг/м}^3$ .

$$L = \frac{(1,715 \cdot 860)}{0,24 \cdot 6 \cdot 1,206} = 849,1 \text{ м}^3/\text{ч}.$$

#### 5.2.4 Выбор кондиционера

Исходя из полученных данных, выберем инверторную сплит-систему LG S12SWC, который будет удовлетворять полученным требованиям по необходимому количеству воздуха.

Таблица 5.1 – Основные технические характеристики кондиционера LG S12SWC.

Мощность охл./ обогрев, кВт	Потребляемая мощность охл. / обогрев, кВт	Площадь помещения, м <sup>2</sup>	Расход воздуха, м <sup>3</sup> /ч
3,6/3,8	0,82/0,87	35	870

Возможности:

- охлаждение/обогрев/вентиляция/автоматический режим;
- экономичный режим;
- дезодорирующий фильтр;
- антикоррозийное покрытие Gold Fin;
- оптимизированная подача воздуха;
- регулируемые жалюзи;
- режим самодиагностики;
- 24 часовой таймер;
- автоматический перезапуск;

- моющаяся передняя панель;
- новый многофункциональный пульт дистанционного управления;
- хладагент — R410A.

## Заключение

В дипломном проекте была разработана автоматизированная информационная система «Vappetit» для кафе «VonAppetit». Рассмотрены проблемы автоматизации деятельности предприятий общественного питания. Обоснована необходимость разработки АИС «Vappetit», которая позволяет повысить эффективность работы заведения.

Первая глава посвящена изучению предметной области исследования. Информационные технологии в ресторанном бизнесе являются эффективным инструментом управления и учета, а также привлечения клиентов. Это обусловлено тем, что любое заведение, будь-то небольшое кафе или крупная ресторанная сеть, нуждается в системах, контролирующих деятельность заведения и позволяющих быстро эффективно осуществлять управление им.

Вторая глава посвящена разработке UML-диаграмм и проектированию базы данных АИС «Vappetit». Для построения моделей использовались UML-диаграммы и методология IDEF1X.

Интерфейс системы имеет удобный и дружелюбный вид. Используемый стиль упрощает работу с программой. Рабочая система содержит следующие окна и панели: окно авторизации, окно меню, окно администратора, панели выбранных блюд, материалов, добавок, учета скидки и т.д.

Четвертая глава посвящена технико-экономическому расчету. В результате экономического расчета затраты на разработку программного продукта для кафе «Vappetit» составили 856 539 тенге. Основной статьей расходов является заработная плата, которая составляет 83,3% от всех затрат. На втором месте прочие расходы, которые составляют 14,4% от всех затрат. Цена реализации программного продукта - 1 199 155 тенге.

В пятой главе был произведен анализ условий труда в кафе «VonAppetit» и выполнен расчет систем кондиционирования. Из расчетов следует отметить, что при достаточно маленьком пространстве и большом количестве человек и оборудования, количество избыточного тепла очень высоко, что предполагает установку достаточно мощной системы кондиционирования.

Исходя из полученных данных, была выбрана инверторная сплит-система LG S12SWC, которая будет удовлетворять полученным требованиям по необходимому количеству воздуха в помещении.

## Список литературы

- 1 Карачаровский В. ИКТ в ресторанном бизнесе – насущная потребность или опережение времени [Электронный ресурс]. – Режим доступа: [http://www.cnews.ru/reviews/free/trade2008/articles/restaurant\\_3.shtml](http://www.cnews.ru/reviews/free/trade2008/articles/restaurant_3.shtml).
- 2 Сайт о бизнесе [Электронный ресурс]. – Режим доступа: <http://kommersant.uz/>
- 3 Библиотека MNWhoost.ru [Электронный ресурс] – Режим доступа: [http://sitemonitor.ru/doc/UML\\_HTM](http://sitemonitor.ru/doc/UML_HTM)
- 4 У. Боггс, М. Боггс «UML и Rational Rose 2002» - Издательство «ЛЮРИ», 2004.;
- 5 Диаграммы UML [Электронный ресурс] – Режим доступа: <http://www.intuit.ru/studies/courses>
- 6 Бекишева А. И. Методические указания к выполнению экономической части дипломной работы для бакалавров специальности 5В070300 – Информационные системы – Алматы: АУЭС; 2013. – 24с.
- 7 Хакимжанов Т. И. Методические указания к выполнению раздела в дипломных проектах(для студентов всех форм обучения всех специальностей) – Алматы: АИЭС; 2002. – 29с.
- 8 Основы UML – Разработка диаграмм в среде Rational Rose[Электронный ресурс] – Режим доступа: <http://2programmer.ru/uml/>
- 9 Л. Веллинг, Л. Томсон. Разработка веб-приложений с помощью PHP и MySQL, 4-е изд.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2010. – 848 с.: ил. – Парал. Тит. Англ.
- 10 Э. Харрис. PHP/MySQL для начинающих. /Пер. с англ – М КУДИЦ-ОБРАЗ, 2005 – 384 с.
- 11 В. А. Камаев, В. В. Костерин. Технологии программирования, 2-е изд. «Высшая школа», 2006. – 454 с.
- 12 С. Маклаков. BPwin и Erwin. CASE-средства для разработки информационных систем – «Диалог-МИФИ», 2000 – 631 с.
- 13 Знакомство с Kohana 3.0 – Часть 3 [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/111537>
- 14 Задачи автоматизации ресторана, кафе, бара [Электронный ресурс]. – Режим доступа: [http://club.cnews.ru/blogs/entry/zadachi\\_avtomatizatsii\\_restorana\\_kafe\\_bara](http://club.cnews.ru/blogs/entry/zadachi_avtomatizatsii_restorana_kafe_bara)
- 15 Проектирование информационных систем [Электронный ресурс]. – Режим доступа: <http://lib.convdocs.org/docs/index-190199.html?page=2>

16 Общие положения и требования к разрабатываемым системам  
[Электронный ресурс] – Режим доступа:  
<http://www.pandia.ru/text/78/473/32201.php>

## **Приложение А**

### **Техническое задание**

#### **А.1 Общие положения**

##### **А.1.1 Полное название системы**

Полное наименование системы: Автоматизированная информационная система «BonAppetit». Краткое наименование системы: АИС «BonAppetit»

##### **А.1.2 Шифр темы и номер контракта**

Шифр темы: АИС-BonAppetit-14.

Номер контракта: №1/01-09-13-001 от 02.09.2013 г.

##### **А.1.3 Наименование организаций заказчика и разработчика**

Заказчиком системы является кафе «BonAppetit», г. Семей.

Адрес заказчика: 050013, Республика Казахстан, г. Семей, ул. Интернациональная 31.

Разработчиком системы является, Токтаров Бауыржан Токтарулы, студент АУЭС, ФИТ, группа ИС 10-2.

Адрес разработчика: г. Алматы, ул. Шашкина 14, к. 415.

##### **А.1.4 Документы, на базе которых разрабатывается система, кем и когда утверждены эти документы**

Основанием для разработки АИС «BonAppetit» являются следующие документы и нормативные акты:

Контракт №1/01-09-13-001 от 02.09.2013 года на выполнение работ по выполнению первого этапа работ по созданию Автоматизированная информационная система «BonAppetit»;

Договор разработчика с заказчиком о выполнении работы по созданию системы в указанные сроки, утвержденные директором кафе «BonAppetit» Жоламановым Н. А.;

Контракт;

На основе задания на дипломный проект.



### *Продолжение приложения А*

#### А.1.5 Запланированные даты начала и окончания работ по разработке системы

Плановый срок начала работ по созданию АИС «BonAppetit» - 02 сентября 2013 года.

Плановый срок окончания работ по созданию АИС «BonAppetit»- 20 мая 2014 года.

#### А.1.6 Информация об источниках и процедуре финансирования

Источником финансирования является бюджет кафе «BonAppetit».

Процедура финансирования определяется условиями директора кафе «BonAppetit».

#### А.1.7 Порядок оформления и предоставления заказчику результатов работ

Система передается в виде функционирующего комплекса на базе средств вычислительной техники заказчика в сроки, установленные заказчиком. Приемка системы осуществляется комиссией в составе уполномоченных представителей заказчика.

Порядок предъявления системы, ее испытаний и окончательной приемки определен в п.6 настоящего ТЗ. Совместно с предъявлением системы производится сдача разработанного исполнителем комплекта документации согласно п.8 настоящего ТЗ.

#### А.1.8 Состав используемой нормативно-технической документации

При разработке автоматизированной управляющей системы и создании проектно-эксплуатационной документации Исполнитель должен руководствоваться требованиями следующих нормативных документов:

ГОСТ 34.601-90. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания;

ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплексность и обозначение документов при создании автоматизированных систем;

РД 50-34.698-90. Методические указания. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов.

### *Продолжение приложения А*

#### **А.2 Назначение и цели создания (развития) системы**

– назначение системы – проектируемая система предназначена для использования её сотрудниками кафе, т.е официантом и администратором. Её назначение – автоматизация деятельности кафе.

– цели создания системы – повышение эффективности работы заведения за счет оперативного контроля заказов и создания систематизированной базы данных.

##### **А.2.1 Назначение АИС**

АИС "Vappetit" предназначена для комплексного информационно-аналитического обеспечения процессов деятельности кафе «BonAppetit», в части исполнения следующих процессов:

– хранение информации в базе данных с обеспечением удобного, быстрого поиска необходимых данных и надежной системы разграничения прав доступа;

– резервное дублирование(бэкап) данных;

– организация оперативной деятельности по обработке заказов;

– обеспечение и контроль дисконтной политики;

– обеспечение доступа администратора к информации, имеющей конфиденциальный характер;

– организация эффективного управления и реагирования на происходящие бизнес-процессы;

##### **А.2.2 Основные цели создания АИС «Vappetit»**

– решение регламентных задач финансового учета заказов и материалов;

– снижение временных затрат на обработку заказов;

– получение отчетов о выполненных операциях, используя созданную базу данных;

– ведение и обновление количественно-суммовой статистики;

– повышение эффективности и оперативности работы сотрудников кафе;

Для реализации поставленных целей система должна решать следующие задачи:

- распределение прав доступа;
- автоматизация финансового учета заказов и материалов;
- реализация количественно-суммовой статистики;
- введение контролируемой дисконтной политики;
- автоматизация оформления заказов;
- калькуляция заказов;

*Продолжение приложения А*

- активация текущих заказов;
- учёт постоянных ассортиментов;
- безопасность информации коммерческого характера;
- обеспечение защиты от несанкционированного доступа и искажения или удаления информации;
- оперативная обработка и активация заказов;

### **А.3 Характеристика объекта автоматизации**

#### **А.3.1 Объект автоматизации**

Процессы по управлению АИС "BonAppetit", а также контроль эффективности выполнения указанных процессов. Данные процессы осуществляются следующими специалистами:

- персонал кафе.

#### **А.3.2 Существующее программное обеспечение**

В настоящий момент деятельность кафе «BonAppetit» не автоматизирована.

#### **А.3.3 Существующее нормативно-правовое обеспечение**

Существующее нормативно-правовое обеспечение составляют федеральные и областные нормативные правовые акты:

- Конституция РК;
- Гражданский кодекс РК и т.д.

### **А.4 Требования к системе**

Требования к системе в целом:

- требования к структуре и функционированию системы;
- требования к персоналу системы;
- показатели назначения;
- требования к надежности; безопасности; эргономике и технической эстетике; эксплуатации, техническому обслуживанию, ремонту и хранению

компонентов системы; защите информации; сохранности информации при авариях; защите от влияния внешних воздействий;

– требования к патентной чистоте; стандартизации и унификации; дополнительные требования.

Требования к функциям (задачам), выполняемым системой; перечень функций, задач или их комплексов, подлежащих автоматизации (по каждой подсистеме); очередность ввода в эксплуатацию, временной регламент

#### *Продолжение приложения А*

реализации и требования к качеству реализации каждой функции, задачи (или комплекса задач), к форме представления выходной информации, перечень и критерии отказов для каждой функции, по которой задаются требования по надежности.

Требования к видам обеспечения в зависимости от вида системы приводят требования к математическому, информационному, лингвистическому, программному, техническому, метрологическому, организационному, методическому и другим видам обеспечения системы.

#### А.4.1 Требования к системе в целом

##### А.4.1.1 Требования к структуре и функционированию системы

А.4.1.1.1 В составе АИС "Vappetit" должны решаться следующие задачи:

- резервное дублирование информации, содержащейся в БД;
- электронное меню и обработка заказа;
- формирование отчетности;
- система хранения информации;

Подсистема резервного дублирования информации подразумевает обеспечение целостности и сохранение одной из главных ценностей системы – накопленная база данных, сбой которой приведет непоправимым последствиям для заведения. Поэтому необходимо исключить вероятность потери и порчи данных, при том, что для больших нагруженных БД при ошибке сервера эта вероятность близка к 100%.

Электронное меню позволяет оперативно управлять процессом выбора ассортимента. Выполнение таких стандартных операций, как выбор блюда, подбор добавок и учет скидки, требует от исполнителя всего лишь нескольких щелчков мышью. Благодаря возможности автоматического расчета дисконта для заказа, исключается вероятность неверного расчета суммы заказа.

В любой момент принятия заказа можно внести необходимые изменения в ассортимент и добавки, или вернувшись в меню полностью поменять заказ. Эта возможность исключает наличие несуществующих заказов в БД.

Под видами автоматизированной обработки считаются процедуры, выполняемые с заказом с целью его изменения. К таким процедурам можно отнести:

Общие виды обработки:

- создание;
- редактирование;
- сохранение;
- активация;
- выбор по добавкам;
- выбор по материалам;
- перемещение в меню текущих заказов;

#### *Продолжение приложения А*

Электронная обработка заказа осуществляется в рамках бизнес-процесса, предусмотренного правами доступа, которые предоставлены сотрудникам, являющимся пользователями системы.

Говоря о порядке обработки заказов, можно выделить несколько этапов:

- прием и первичный выбор блюд и напитков;
- рассмотрение заказа в окне «корзины», выбор добавок, материалов и учет скидки;
- перемещение заказа в панель текущих заказов;
- активация заказа.

Подсистема формирования отчетности предназначена для создания и ведения количественно-суммовой статистики в виде, удобном для вывода на печатающие устройства на основе данных АИС «Vappetit», редактирование и изменение дисконтов по позициям меню, просмотр статистики закупок по блюдам, формирования и предоставления аналитических и статистических отчетов в различных форматах (включая графические), отображения регламентированных отчетов с помощью web-интерфейса, вывода подготовленных отчетных форм на печать.

Подсистема хранения информации. Заказ состоит из набора атрибутов, определяющие его свойства и используемые для различных автоматизированных задач, таких как: обработка, классификация, построение отчетов, групповая обработка, и т.д., и непосредственно файлов с информацией. Вся необходимая информация, как правило, хранятся в таблицах базы данных. Все данные по принятым заказам и прочим операциям заносятся в БД. Доступ к такого рода информации, имеющей конфиденциальный характер, предоставляется только администратору. Прочий доступ к данным запрещен.

АИС «Vappetit» должна обеспечивать доступ администратору кафе к операционным данным БД и АИС «Vappetit» (путем предоставления сервисов, позволяющих формировать запросы на получение информации о деятельности кафе в реальном времени).

**А.4.1.1.2 Требования к методам и инструментам связи для информационного взаимодействия между элементами системы системы**

Входящие в состав АИС «Vappetit» подсистемы в процессе работы должны обмениваться информацией на базе открытых способов обмена данными, используя для этого включенные в них модули информационного взаимодействия.

Форматы данных будут разработаны и утверждены на этапе технического проектирования.

#### *Продолжение приложения А*

##### А.4.1.1.3 Требования к режимам деятельности системы

Для АИС «Vappetit» определены следующие режимы функционирования:

- нормальный режим функционирования;
- аварийный режим функционирования.

Основным режимом функционирования АС является нормальный режим.

В нормальном режиме функционирования системы:

- клиентское программное обеспечение и технические средства пользователей и администратора системы обеспечивают возможность функционирования в течение рабочего дня (с 09:00 до 17:00) шесть дней в неделю;
- серверное программное обеспечение и технические средства серверов обеспечивают возможность круглосуточного функционирования, с перерывами на обслуживание;
- исправно работает оборудование, составляющее комплекс технических средств;
- исправно функционирует системное, базовое и прикладное программное обеспечение системы.

Для обеспечения нормального режима функционирования системы необходимо выполнять требования и выдерживать условия эксплуатации программного обеспечения и комплекса технических средств системы, указанные в соответствующих технических документах (техническая документация, инструкции по эксплуатации и т.д.).

Аварийный режим функционирования системы характеризуется отказом одного или нескольких компонент программного и (или) технического обеспечения.

В случае перехода системы в аварийный режим необходимо:

- завершить работу всех приложений, с сохранением данных;
- выключить рабочие станции операторов;
- выключить все периферийные устройства;
- выполнить резервное копирование БД.

После этого необходимо выполнить комплекс мероприятий по устранению причины перехода системы в аварийный режим.

#### А.4.1.1.4 Требования по диагностированию системы

АИС «Vарретит» должна предоставлять инструменты диагностирования основных процессов системы, трассировки и мониторинга процесса выполнения программы.

#### *Продолжение приложения А*

Компоненты должны предоставлять удобный интерфейс для возможности просмотра диагностических событий, мониторинга процесса выполнения программ.

При возникновении аварийных ситуаций, либо ошибок в программном обеспечении, диагностические инструменты должны позволять сохранять полный набор информации, необходимой разработчику для идентификации проблемы (снимки экранов, текущее состояние памяти, файловой системы).

#### А.4.1.1.5 Перспективы развития и совершенствования системы

АИС «Vарретит» должна реализовывать возможность дальнейшей модернизации как программного обеспечения.

#### А.4.1.2 Требования к количеству и компитентности персонала системы

Для эксплуатации АИС «Vарретит» определены следующие роли:

- системный администратор;
- администратор баз данных;
- администратор информационной безопасности;
- пользователь.

Основными обязанностями системного администратора являются:

- модернизация, настройка и мониторинг работоспособности комплекса технических средств (серверов, рабочих станций);
- установка, модернизация, настройка и мониторинг работоспособности системного и базового программного обеспечения;
- установка, настройка и мониторинг прикладного программного обеспечения;
- ведение учетных записей пользователей системы.
- системный администратор должен обладать высоким уровнем квалификации и практическим опытом выполнения работ по установке, настройке и администрированию программных и технических средств, применяемых в системе.

Основными обязанностями администратора баз данных являются:

- установка, модернизация, настройка параметров программного обеспечения СУБД;
- оптимизация прикладных баз данных по времени отклика, скорости доступа к данным;
- разработка, управление и реализация эффективной политики доступа к информации, хранящейся в прикладных базах данных.
- администратор баз данных должен обладать высоким уровнем квалификации и практическим опытом выполнения работ по установке, настройке и администрированию используемых в АС СУБД.

#### *Продолжение приложения А*

Основными обязанностями администратора информационной безопасности являются:

- разработка, управление и реализация эффективной политики информационной безопасности системы;
- управление правами доступа пользователей к функциям системы;
- осуществление мониторинга информационной безопасности.

Администратор информационной безопасности данных должен обладать высоким уровнем квалификации и практическим опытом выполнения работ по обеспечению информационной безопасности.

Пользователи системы должны иметь опыт работы с персональным компьютером на базе операционных систем Microsoft Windows на уровне квалифицированного пользователя и свободно осуществлять базовые операции в стандартных Windows.

Роли системного администратора, администратора баз данных и администратора информационной безопасности могут быть совмещены в роль.

Рекомендуемая численность для эксплуатации АИС «Vappetit»:

Администратор - 1 штатная единица;

Пользователь - число штатных единиц определяется структурой заведения.

#### А.4.1.3 Параметры назначения

АИС «Vappetit» должны обеспечивать возможность исторического хранения данных с глубиной не менее 10 лет.

Система должна обеспечивать возможность одновременной работы 50 пользователей для подсистемы операционной деятельности, и не менее 10-ти пользователей для других подсистем при следующих характеристиках времени отклика системы:

- для операций навигации по экранным формам системы - не более 5 сек;
- для операций формирования справок и выписок - не более 10 сек.



Время формирования аналитических отчетов определяется их сложностью и может занимать продолжительное время.

Система должна предусматривать возможность масштабирования по производительности и объему обрабатываемой информации без модификации ее программного обеспечения путем модернизации используемых технических средств.

#### *Продолжение приложения А*

##### А.4.1.4 Требования к надежности

Система должна сохранять работоспособность и обеспечивать восстановление своих функций при возникновении следующих внештатных ситуаций:

- при сбоях в системе электроснабжения аппаратной части, приводящих к перезагрузке ОС, восстановление программы должно происходить после перезапуска ОС и запуска исполняемого файла системы;

- при ошибках в работе аппаратных средств (кроме носителей данных и программ) восстановление функции системы возлагается на ОС;

- при ошибках, связанных с программным обеспечением (ОС и драйверы устройств), восстановление работоспособности возлагается на ОС.

Для защиты аппаратуры от бросков напряжения и коммутационных помех должны применяться сетевые фильтры.

##### А.4.1.5 Требования к безопасности

Все внешние элементы технических средств системы, находящиеся под напряжением, должны иметь защиту от случайного прикосновения, а сами технические средства иметь зануление или защитное заземление в соответствии с ГОСТ 12.1.030-81.

Система электропитания должна обеспечивать защитное отключение при перегрузках и коротких замыканиях в цепях нагрузки, а также аварийное ручное отключение.

Общие требования пожарной безопасности должны соответствовать нормам на бытовое электрооборудование. В случае возгорания не должно выделяться ядовитых газов и дымов. После снятия электропитания должно быть допустимо применение любых средств пожаротушения.

Факторы, оказывающие вредные воздействия на здоровье со стороны всех элементов системы (в том числе инфракрасное, ультрафиолетовое, рентгеновское и электромагнитное излучения, вибрация, шум,

электростатические поля, ультразвук строчной частоты и т.д.), не должны превышать действующих норм.

#### А.4.1.6. Требования к эргономике и технической эстетике

Взаимодействие пользователей с прикладным программным обеспечением, входящим в состав системы должно осуществляться посредством визуального графического интерфейса (GUI). Интерфейс системы должен быть понятным и удобным, не должен быть перегружен графическими элементами и должен обеспечивать быстрое отображение экранных форм. Навигационные элементы должны быть выполнены в

##### *Продолжение приложения А*

удобной для пользователя форме. Средства редактирования информации должны удовлетворять принятым соглашениям в части использования функциональных клавиш, режимов работы, поиска, использования оконной системы. Ввод-вывод данных системы, прием управляющих команд и отображение результатов их исполнения должны выполняться в интерактивном режиме. Интерфейс должен соответствовать современным эргономическим требованиям и обеспечивать удобный доступ к основным функциям и операциям системы.

Управление системой должно осуществляться с помощью набора экранных меню, кнопок, значков и т. п. элементов. Клавиатурный режим ввода должен использоваться главным образом при заполнении и/или редактировании текстовых и числовых полей экранных форм.

Система должна обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных. В указанных случаях система должна выдавать пользователю соответствующие сообщения, после чего возвращаться в рабочее состояние, предшествовавшее неверной (недопустимой) команде или некорректному вводу данных.

Система должна соответствовать требованиям эргономики и профессиональной медицины при условии комплектования высококачественным оборудованием (ПЭВМ, монитор и прочее оборудование), имеющим необходимые сертификаты соответствия и безопасности.

#### А.4.1.7 Требования к использованию, техническому обеспечению, наладке и хранению элементов системы

Техническая и физическая защита аппаратных компонентов системы, носителей данных, бесперебойное энергоснабжение, резервирование ресурсов, текущее обслуживание реализуется техническими и

организационными средствами, предусмотренными в ИТ инфраструктуре Заказчика.

Для нормальной эксплуатации разрабатываемой системы должно быть обеспечено бесперебойное питание ПК. При эксплуатации система должна быть обеспечена соответствующая стандартам хранения носителей и эксплуатации ПК температура и влажность воздуха.

Периодическое техническое обслуживание используемого ПК должно проводиться в соответствии с требованиями технической документации изготовителей, но не реже одного раза в год.

Периодическое техническое обслуживание и тестирование ПК должны включать в себя обслуживание и тестирование ПК, кабельной системы, устройств бесперебойного питания.

#### *Продолжение приложения А*

В процессе проведения периодического технического обслуживания должны проводиться внешний и внутренний осмотр и чистка ПК, проверка контактных соединений, проверка параметров настроек работоспособности ПК.

Размещение ПК должно исключать возможность бесконтрольного проникновения в них посторонних лиц и обеспечивать сохранность находящихся в этих помещениях конфиденциальных документов и технических средств.

Размещение ПК должно соответствовать требованиям техники безопасности, санитарным нормам и требованиям пожарной безопасности.

Все пользователи системы должны соблюдать правила эксплуатации электронной вычислительной техники.

Квалификация персонала и его подготовка должны соответствовать технической документации.

#### **А.4.1.8 Требования к обеспечению защиты информации от несанкционированного доступа**

АИС должна обеспечивать защиту от несанкционированного доступа (НСД).

Компоненты подсистемы защиты от НСД должны обеспечивать:

- идентификацию пользователя;
- проверку полномочий пользователя при работе с системой;
- разграничение доступа пользователей на уровне задач и информационных массивов.

Протоколы аудита системы и приложений должны быть защищены от несанкционированного доступа как локально, так и в архиве.

Защищённая часть системы должна использовать "слепые" пароли (при наборе пароля его символы не показываются на экране либо заменяются одним типом символов; количество символов не соответствует длине пароля).

Защищённая часть системы должна автоматически блокировать сессии пользователей и приложений по заранее заданным временам отсутствия активности со стороны пользователей и приложений.

Защищённая часть системы должна использовать многоуровневую систему защиты. Защищённая часть системы должна быть отделена от незащищённой части системы межсетевым экраном.

#### А.4.1.9 Требования по безопасности информации при авариях

Программное обеспечение АИС «Vappetit» должно восстанавливать свое функционирование при корректном перезапуске аппаратных средств. Должна быть предусмотрена возможность организации автоматического и

##### *Продолжение приложения А*

(или) ручного резервного копирования данных системы средствами системного и базового программного обеспечения (ОС, СУБД), входящего в состав программно технического комплекса Заказчика.

Приведенные выше требования не распространяются на компоненты системы, разработанные третьими сторонами и действительны только при соблюдении правил эксплуатации этих компонентов, включая своевременную установку обновлений, рекомендованных производителями покупного программного обеспечения.

#### А.4.1.10 Требования к патентной чистоте

Установка системы в целом, как и установка отдельных частей системы не должна предъявлять дополнительных требований к покупке лицензий на программное обеспечение сторонних производителей.

#### А.4.1.11 Требования по стандартизации и унификации

Экранные формы должны проектироваться с учетом требований унификации:

- все экранные формы пользовательского интерфейса должны быть выполнены в едином графическом дизайне, с одинаковым расположением основных элементов управления и навигации;

- для обозначения сходных операций должны использоваться сходные графические значки, кнопки и другие управляющие (навигационные) элементы;

- термины, используемые для обозначения типовых операций (добавление информационной сущности, редактирование поля данных), а также последовательности действий пользователя при их выполнении, должны быть унифицированы;

– внешнее поведение сходных элементов интерфейса (реакция на наведение указателя "мыши", переключение фокуса, нажатие кнопки) должны реализовываться одинаково для однотипных элементов.

#### А.4.2 Требования к видам обеспечения

##### А.4.2.1 Требования к математическому обеспечению системы

Математические методы и алгоритмы, используемые для шифрования/дешифрования данных, а также программное обеспечение, реализующее их, должны быть сертифицированы уполномоченными организациями.

#### *Продолжение приложения А*

##### А.4.2.2 Требования к информационному обеспечению системы

Состав, структура и способы организации данных в системе должны быть определены на этапе технического проектирования.

Хранение данных должно осуществляться на основе современных реляционных или СУБД. Для обеспечения целостности данных должны использоваться встроенные механизмы СУБД.

Средства СУБД, а также средства используемых операционных систем должны обеспечивать документирование и протоколирование обрабатываемой в системе информации.

Структура базы данных должна поддерживать кодирование хранимой и обрабатываемой информации в соответствии с общепринятыми классификаторами (там, где они применимы).

Доступ к данным должен быть предоставлен только авторизованным пользователям с учетом их служебных полномочий, а также с учетом категории запрашиваемой информации.

Структура базы данных должна быть организована рациональным способом, исключающим единовременную полную выгрузку информации, содержащейся в базе данных системы.

Технические средства, обеспечивающие хранение информации, должны использовать современные технологии, позволяющие обеспечить повышенную надежность хранения данных и оперативную замену оборудования (распределенная избыточная запись/считывание данных; зеркалирование; независимые дисковые массивы; кластеризация).

В состав системы должна входить специализированная подсистема резервного копирования и восстановления данных.

При проектировании и развертывании системы необходимо рассмотреть возможность использования накопленной информации из уже функционирующих информационных систем.

#### А.4.2.3 Требования к лингвистическому обеспечению системы

Все прикладное программное обеспечение системы для организации взаимодействия с пользователем должно использовать русский язык.

#### А.4.2.4 Требования к программному обеспечению системы

При проектировании и разработке системы необходимо максимально эффективным образом использовать ранее закупленное программное обеспечение, как серверное, так и для рабочих станций.

Используемое при разработке программное обеспечение и библиотеки программных кодов должны иметь широкое распространение, быть

*Продолжение приложения А*

общедоступными и использоваться в промышленных масштабах. Базовой программной платформой должна являться операционная система MS Windows.

#### А.4.2.5 Требования к техническому обеспечению

Техническое обеспечение системы должно максимально и наиболее эффективным образом использовать существующие в органах федерального агентства технические средства.

В состав комплекса должны следующие технические средства:

Серверы БД;

Серверы приложений;

Сервер системы формирования отчетности;

Web сервер;

ПК пользователей;

#### А.4.2.6 Требования к организационному обеспечению

Организационное обеспечение системы должно быть достаточным для эффективного выполнения персоналом возложенных на него обязанностей при осуществлении автоматизированных и связанных с ними неавтоматизированных функций системы.

Заказчиком должны быть определены должностные лица, ответственные за:

– обработку информации АС;

– администрирование АС;

– обеспечение безопасности информации АС;

– управление работой персонала по обслуживанию АС.

К работе с системой должны допускаться сотрудники, имеющие навыки работы на персональном компьютере, ознакомленные с правилами эксплуатации и прошедшие обучение работе с системой.

#### А.4.2.7 Требования к методическому обеспечению

В состав методического обеспечения системы должны входить законодательные акты, стандарты, нормативы, инструкции.

### *Продолжение приложения А*

## **А.5 Порядок контроля и приемки системы**

### А.5.1 Виды, состав, объем и методы испытаний системы

Виды, состав, объем, и методы испытаний подсистемы должны быть изложены в программе и методике испытаний АИС «Vappetit», разрабатываемой в составе рабочей документации.

### А.5.2 Общие требования к приемке работ по стадиям

Сдача-приёмка работ производится поэтапно, в соответствии с рабочей программой и календарным планом.

Сдача-приемка осуществляется комиссией, в состав которой входят представители Заказчика и Исполнителя.

Все создаваемые в рамках настоящей работы программные изделия (за исключением покупных) передаются Заказчику, как в виде готовых модулей, так и в виде исходных кодов, представляемых в электронной форме на стандартном машинном носителе (например, на компакт-диске).

### А.5.3 Статус приемочной комиссии

Статус приемочной комиссии определяется Заказчиком до проведения испытаний.

## **А.6 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие**

В ходе выполнения проекта на объекте автоматизации требуется выполнить работы по подготовке к вводу системы в действие. При подготовке

к вводу в эксплуатацию АИС «Vappetit» Заказчик должен обеспечить выполнение следующих работ:

Определить ответственных должностных лиц, ответственных за внедрение и проведение опытной эксплуатации АИС «Vappetit»;

Обеспечить присутствие пользователей на обучении работе с системой, проводимом Исполнителем;

Совместно с Исполнителем подготовить план развертывания системы на технических средствах Заказчика;

Провести опытную эксплуатацию АИС «Vappetit».

### *Продолжение приложения А*

#### **А.7 Требования к документированию**

Данный проект сопровождается ТЗ и документацией на технический проект.

В техническом задании расписываются:

- основные цели, задачи, сроки и этапы разработки;
- список основных функций и требований;
- список функций интерфейса.

Документация на технический проект является инструкцией по использованию данного ПО. В данной документации будут расписываться:

- условия работы ПО;
- установка ПО;
- использование ПО, попунктное описание основных функций и функций интерфейса;
- устранение проблем при переходе в аварийный режим.

Для системы на различных стадиях создания должны быть выпущены следующие документы из числа предусмотренных в ГОСТ 34.201-«Информационная технология. Комплекс стандартов на автоматизированные системы».

#### **А.8 Источники разработки**

Документы и информационные материалы (технико-экономическое обоснование, отчеты о законченных научно-исследовательских работах, информационные материалы на отечественные, зарубежные системы-аналоги и др.), на основании которых разрабатывалось ТЗ и которые должны быть использованы при создании системы.

Технико-экономическое обоснование. Данный документ содержит, финансовое описание системы, в котором содержится перечень используемых



ресурсов и их ценовое описание. Стоимость системы, подсчет рентабельности. Минимизация используемых ресурсов для получения наиболее максимальной прибыли.

Данная система должна разрабатываться на основании ТК 34 по стандартизации Информационные технологии. Номер приказа и дата утверждения: от 27.07.01 г. № 274.

## **Приложение Б**

### **Листинг программы**

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Controller_Admin_Admintemplate extends Controller_Template {
```

```
    public $template = 'admin_new/admintemplate';
```

```
    public function before() {
```

```
        parent::before();
```

```
        if(Auth::instance()->logged_in('admin')) {
```

```
            $this->template->title = ";
```

```
            $this->template->content = ";
```

```
            $this->template->styles = array();
```

```
            $this->template->scripts = array();
```

```
            $styles = array(
```

```
                '/files/admin/css/main_admin.css' => 'screen',
```

```
            );
```

```
            $scripts = array(
```

```
                '/files/admin/js/dropdown.js',
```

```
                '/files/admin/js/jquery-1.8.js',
```

```
            );
```

```

        $this->template->styles = array_merge( $this->template->styles, $styles );
        $this->template->scripts = array_merge( $this->template->scripts,
$scripts );

        } else {
            Controller::redirect('login');
        }
    } //End Function before

} //End Template_Controller Admintemplate
<?php defined('SYSPATH') or die('No direct script access.');
```

```

class Controller_Admin_Edit extends Controller_Admin_Admintemplate {
    Продолжение приложения Б
```

```

    /*
    * Редактирование цен и скидок
    * 1. Блюды
    * 2. Напитков
    */
    public function action_dishes() {
        $this->template->title = 'Редактирование Цен и Скидок';

        $dish = ORM::factory('Dish');
        $prices_dishes = $dish->get_menu('блюды');

        $content = View::factory('admin_new/edit/menu')
            ->bind('prices_dishes', $prices_dishes);
        $this->template->content = $content;

    } //End Action dishes

    public function action_drinks() {
        $this->template->title = 'Редактирование Цен и Скидок';

        $dish = ORM::factory('Dish');
        $prices_dishes = $dish->get_menu('напиток');

        $content = View::factory('admin_new/edit/menu')
            ->bind('prices_dishes', $prices_dishes);
        $this->template->content = $content;

    } //End Action dishes
```

```

public function action_menu() {
    $this->template->title = 'Редактирование Цен и Скидок';

    $dish = ORM::factory('Dish');
    $prices_dishes = $dish->get_menu();

    $content = View::factory('admin_new/edit/menu')
        ->bind('prices_dishes', $prices_dishes);
    $this->template->content = $content;

} //End Action menu

public function action_orders() {
    Продолжение приложения Б
    $this->template->title = 'Редактирование заказов';

    $data = $this->request->post();

    if(isset($data['order_id'])) {

        $found_order = ORM::factory('Order')
            ->where('id', '=', $data['order_id'])
            ->find();
        $order_products = $found_order->product->find_all();

        $content = View::factory('admin_new/edit/orders')
            ->bind('found_order', $found_order)
            ->bind('order_products', $order_products);
    } else {
        $content = View::factory('admin_new/edit/orders');
    }

    $this->template->content = $content;

} //End Action dishes

public function action_juices() {
    $this->template->title = 'Редактирование напитков';

    $other_production = ORM::factory('Dish')->get_menu('сок');

    $product_type = "сок";
    $product_image = "drinks";

```

```
$content = View::factory('admin_new/edit/other')
    ->bind('other_production', $other_production)
    ->bind('product_type', $product_type)
    ->bind('product_image', $product_image);
```

```
$this->template->content = $content;
```

```
}//End Action juice
```

```
public function action_chocolates() {
    $this->template->title = 'Редактирование шоколадок';
```

*Продолжение приложения Б*

```
$other_production = ORM::factory('Dish')->get_menu('шоколад');
```

```
$product_type = "шоколад";
$product_image = "chocolates";
```

```
$content = View::factory('admin_new/edit/other')
    ->bind('other_production', $other_production)
    ->bind('product_type', $product_type)
    ->bind('product_image', $product_image);
```

```
$this->template->content = $content;
```

```
}//End Action juice
```

```
}//End Controller Admin_Edit
```

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Controller_Admin_Orders extends Controller_Admin_Admintemplate {
```

```
/*
```

```
 * Текущие заказы
```

```
*/
```

```
public function action_active() {
```

```
    $this->template->title = 'Текущие заказы';
    $today = date("Y-m-d");
```

```
    $active_orders_s = ORM::factory('Order')
        ->order_by('id', 'DESC')
```

```

        ->limit(20)
        ->where('date', '=', $today)
        ->and_where('status', '=', 'active')
        ->find_all()
        ->as_array();
        $active_orders = array_reverse($active_orders_s);

        $content = View::factory('admin_new/orders/active')
            ->bind('active_orders', $active_orders);
        $this->template->content = $content;

    }//End Action active

```

### *Продолжение приложения Б*

```

/*
 * Последние оформленные
 */
public function action_lasts() {

    $this->template->title = 'Последние оформленные';
    $today = date("Y-m-d");

    $last_orders_s = ORM::factory('Order')
        ->order_by('id', 'DESC')
        ->limit(20)
        ->where('date', '=', $today)
        ->and_where('status', '=', 'complete')
        ->find_all()
        ->as_array();
    $last_orders = array_reverse($last_orders_s);

    $content = View::factory('admin_new/orders/lasts')
        ->bind('last_orders', $last_orders);
    $this->template->content = $content;

} //End Action lasts

} //End Controller Admin_Orders

<?php defined('SYSPATH') or die('No direct script access.');
```

```

class Controller_Admin_Stats extends Controller_Admin_Admintemplate {

```

```

/*
 * Action Сводной статистики
 */
public function action_summary() {
    $this->template->title = 'Суммарная статистика';

    $summary_stats = ORM::factory('Stat')
        ->find_all()
        ->as_array();

    $content = View::factory('admin_new/stats/summary')
        ->bind('summary_stats', $summary_stats);
    $this->template->content = $content;
        Продолжение приложения Б
} //End Action summary

/*
 * Action Статистики по упаковкам
 */
public function action_materials() {
    $this->template->title = 'Статистика по расходным материалам';

    $all_materials = ORM::factory('Material')
        ->find_all()
        ->as_array();

    $content = View::factory('admin_new/stats/materials')
        ->bind('all_materials', $all_materials);
    $this->template->content = $content;

} //End Action packages

public function action_orders() {
    $this->template->title = 'Сводная статистика по заказам';

    $data = $this->request->post();
    if(isset($data['day'])) {
        $day = $data['day'];
    } else {
        $day = date("Y-m-d");
    }

    $all_orders = ORM::factory('Stat', 'orders');

```

```

$day_orders = ORM::factory('Order')
    ->where('date', '=', $day)
    ->find_all();
$day_sum = 0;
foreach($day_orders as $order) {
    $day_sum += $order->sum;
}
$day_orders_count = ORM::factory('Order')
    ->where('date', '=', $day)
    ->count_all();

```

*Продолжение приложения Б*

```

$content = View::factory('admin_new/stats/orders')
    ->bind('day_orders_count', $day_orders_count)
    ->bind('day_sum', $day_sum)
    ->bind('all_orders', $all_orders);
$this->template->content = $content;

```

```

} //End Action packages

```

```

public function action_dishes() {

```

```

    $this->template->title = 'Статистика по блюдам';

```

```

    $data = $this->request->post();

```

```

    if(isset($data['start_day']) && isset($data['end_day'])) {

```

```

        $start_day = $data['start_day'];

```

```

        $end_day = $data['end_day'];

```

```

    } else {

```

```

        $start_day = date("Y-m-") . "01";

```

```

        $end_day = date("Y-m-d");

```

```

    }

```

```

    $order = ORM::factory('Order')->get_maxmin_id($start_day, $end_day);

```

```

    $type = 'блюда';

```

```

    $min_id = $order->id_min;

```

```

    $max_id = $order->id_max;

```

```

    $month_products = ORM::factory('Product')->get_products_stats($min_id,
    $max_id, $type);

```

```

$content = View::factory('admin_new/stats/dishes')
    ->bind('month_products', $month_products)
    ->bind('start_day', $start_day)
    ->bind('end_day', $end_day);
$this->template->content = $content;

} //End Action dishes

public function action_drinks() {
    $this->template->title = 'Статистика по чаю/кофе';

    $data = $this->request->post();
    if(isset($data['start_day']) && isset($data['end_day'])) {
        Продолжение приложения Б
    }

    $start_day = $data['start_day'];
    $end_day = $data['end_day'];
    } else {
    $start_day = date("Y-m-") . "01";
    $end_day = date("Y-m-d");
    }

    $order = ORM::factory('Order')->get_maxmin_id($start_day, $end_day);

    $type = 'напиток';
    $min_id = $order->id_min;
    $max_id = $order->id_max;
    $month_products = ORM::factory('Product')->get_products_stats($min_id,
    $max_id, $type);

    $content = View::factory('admin_new/stats/drinks')
        ->bind('month_products', $month_products)
        ->bind('start_day', $start_day)
        ->bind('end_day', $end_day);
    $this->template->content = $content;

} //End Action drinks

public function action_juices() {
    $this->template->title = 'Статистика по напиткам';

    $data = $this->request->post();
    if(isset($data['start_day']) && isset($data['end_day'])) {
        $start_day = $data['start_day'];

```



```

    $end_day = $data['end_day'];
} else {
    $start_day = date("Y-m-") . "01";
    $end_day = date("Y-m-d");
}

$order = ORM::factory('Order')->get_maxmin_id($start_day, $end_day);

$type = 'сок';
$min_id = $order->id_min;
$max_id = $order->id_max;
$month_products = ORM::factory('Product')->get_products_stats($min_id,
$max_id, $type);

```

*Продолжение приложения Б*

```

$content = View::factory('admin_new/stats/juices')
    ->bind('month_products', $month_products)
    ->bind('start_day', $start_day)
    ->bind('end_day', $end_day);
$this->template->content = $content;

} //End Action juices

public function action_chocolates() {
    $this->template->title = 'Статистика по шоколаду';

    $data = $this->request->post();
    if(isset($data['start_day']) && isset($data['end_day'])) {
        $start_day = $data['start_day'];
        $end_day = $data['end_day'];
    } else {
        $start_day = date("Y-m-") . "01";
        $end_day = date("Y-m-d");
    }

    $order = ORM::factory('Order')->get_maxmin_id($start_day, $end_day);

    $type = 'шоколад';
    $min_id = $order->id_min;
    $max_id = $order->id_max;
    $month_products = ORM::factory('Product')->get_products_stats($min_id,
$max_id, $type);

    $content = View::factory('admin_new/stats/chocolates')

```

```

        ->bind('month_products', $month_products)
        ->bind('start_day', $start_day)
        ->bind('end_day', $end_day);
$this->template->content = $content;

} //End Action drinks

public function action_packages() {
    $this->template->title = 'Расходные материалы за период';

    $data = $this->request->post();
    if(isset($data['start_day']) && isset($data['end_day'])) {
        $start_day = $data['start_day'];
        Продолжение приложения Б
    }
    $end_day = $data['end_day'];
    } else {
        $start_day = date("Y-m-") . "01";
        $end_day = date("Y-m-d");
    }

    $order = ORM::factory('Order')->get_maxmin_id($start_day, $end_day);

    $min_id = $order->id_min;
    $max_id = $order->id_max;
    $month_packages = ORM::factory('Package')->get_packages_stats($min_id,
    $max_id);

    $content = View::factory('admin_new/stats/packages')
        ->bind('month_packages', $month_packages)
        ->bind('start_day', $start_day)
        ->bind('end_day', $end_day);
    $this->template->content = $content;
} //End Action packages

} //End Controller Admin_Stats
<?php defined('SYSPATH') or die('No direct script access.');
```

```

class Controller_Admin_User extends Controller_Admin_Admintemplate {

    /*
    * Action добавления нового кассира/администратора
    */
    public function action_create() {

```

```

$this->template->title = 'Добавление нового Пользователя';

$data = $this->request->post();

/*
 * Если есть данные пытаемся создать пользователя
 */
if(isset($data['username']) && isset($data['password']) &&
isset($data['password_confirm']) && isset($data['email'])) {
    $user = ORM::factory('User');
    $user->values(array(
        'username' => $data['username'],
        Продолжение приложения Б
        'password' => $data['password'],
        'password_confirm' => $data['password_confirm'],
        'email' => $data['email'],
    ));
    $user->save();

    if($user->loaded() && $data['role'] == 'login') {
        $user->add('roles', ORM::factory('Role')->where('name', '=', 'login')-
>find());
    } elseif ($user->loaded() && $data['role'] == 'admin') {
        $user->add('roles', ORM::factory('Role')->where('name', '=', 'login')-
>find());
        $user->add('roles', ORM::factory('Role')->where('name', '=', 'admin')-
>find());
    }

    } else {

    }

    $content = View::factory('admin_new/user/create');
    $this->template->content = $content;

} //End Action create

} //End Controller Admin_Users
<?php defined('SYSPATH') or die('No direct script access.');
```

```

class Controller_Shop_Shop extends Controller_Shop_Shoptemplate {

    /*
    * Меню (основная страница приложения)ы
    */
    public function action_menu() {

        $this->template->title = 'Меню';

        // Инициализируем номера нового заказа и номера продуктов в корзине
        $new_order_id = ORM::factory('Order')->get_order_id();
        $current_product_id = ORM::factory('Product')->get_product_order_id();
        Продолжение приложения Б

        $menu_chocolates = ORM::factory('Dish')->get_menu('шоколад');
        $menu_juices = ORM::factory('Dish')->get_menu('сок');

        //Количество активных заказов на текущий момент
        $active_orders_count = ORM::factory('Order')->get_last_count();
        $current_order_number = ORM::factory('Order')->get_this_number();
        $date = Model::factory('RusDate')->get_now_date();

        if($new_order_id == $current_product_id) {
            $incart_products = ORM::factory('Product')
                ->get_cart_menu($current_product_id);
            $sum = ORM::factory('Product')-
                >get_products_sum($current_product_id);

            $content = View::factory('shop/menu')
                ->bind('menu_chocolates', $menu_chocolates)
                ->bind('menu_juices', $menu_juices)
                ->bind('incart_products', $incart_products)
                ->bind('sum', $sum)
                ->bind('active_orders_count', $active_orders_count)
                ->bind('current_order_number', $current_order_number)
                ->bind('date', $date);
        } else {
            $content = View::factory('shop/menu')
                ->bind('menu_chocolates', $menu_chocolates)
                ->bind('menu_juices', $menu_juices)
                ->bind('active_orders_count', $active_orders_count)
                ->bind('current_order_number', $current_order_number)
                ->bind('date', $date);
        }
    }
}

```

```

}

$this->template->content = $content;
$this->template->styles = array('/files/shop/css/menu_2.css' => 'screen');
$this->template->scripts = array('/files/shop/js/jquery-1.8.js',
                                '/files/shop/js/menu.js');

} //End action_menu

/*
 * Получение содержимого корзины
 */
public function action_cart() {
    Продолжение приложения Б

    // Инициализируем номера сессионного заказа и номера продуктов в
корзине
    $new_order_id = ORM::factory('Order')->get_order_id();
    $product_order_id = ORM::factory('Product')->get_product_order_id();

    if($new_order_id == $product_order_id) {
        $this->template->title = 'Корзина';

        //Получение меню для корзины
        $product = ORM::factory('Product');
        $incart_products = $product->get_cart_menu($new_order_id);

        //Сумма и сумма со скидкой всех продуктов в корзине
        $cart_summ = $product->get_products_sum($new_order_id);
        $sum_with_sale = $product->get_products_sale($new_order_id);

        //Получение материалов для корзины
        $material = ORM::factory('Material');
        $incart_materials = $material->find_all()->as_array();

        $content = View::factory('shop/cart')
            ->bind('incart_products', $incart_products)
            ->bind('incart_materials', $incart_materials)
            ->bind('cart_summ', $cart_summ)
            ->bind('sum_with_sale', $sum_with_sale);
    }
}

```

```

$this->template->content = $content;

//Скрипты и стили для корзины
$this->template->styles = array('/files/shop/css/cart.css' => 'screen');
$this->template->scripts = array('/files/shop/js/jquery-1.8.js',
                                '/files/shop/js/cart.js');

} else {
    Controller::redirect('shop/menu');
}

```

*Продолжение приложения Б*

```

} //End action_cart

/*
 * Оформление заказа
 */
public function action_order() {

    // Инициализируем номера сессионного заказа и номера продуктов в
корзине
    $new_order_id = ORM::factory('Order')->get_order_id();
    $product_order_id = ORM::factory('Product')->get_product_order_id();

    //Сохранение заказа
    if($product_order_id == $new_order_id) {

        //Получение массива
        $post_array = $this->request->post();

        //Дата и время
        $date = date("Y-m-d");
        $time = date("H:i:s");

        //Сумма и сумма со скидкой всех продуктов в корзине
        $sum = ORM::factory('Product')->get_products_sum($new_order_id);
        $sum_with_sale = ORM::factory('Product')-
>get_products_sale($new_order_id);

        if(isset($post_array['with_sale'])) {
            $with_sale = $post_array['with_sale'];

```

```

    } else {
        $with_sale = NULL;
    }

    //Получаем массив с Добавками и добавляем их в Продукты в
корзине
    if(isset($post_array['additions'])) {
        $array_additions = $post_array['additions'];
        ORM::factory('Product')
            ->add_additions($array_additions);
    } else {
        $array_additions = NULL;
        Продолжение приложения Б
    }

    ORM::factory('Product')->change_price($post_array['products'],
$with_sale);

    //Получаем массив с Материалами/упаковками с сохраняем его
    if(isset($post_array['materials'])) {
        $incart_materials = $post_array['materials'];
        $order_packages = ORM::factory('Package');
        $order_packages->save_packages($new_order_id,
$incart_materials);
    } else {
        Controller::redirect('shop/menu');
    }

    //Проверяем заказ на наличие скидки
    if(isset($post_array['with_sale'])) {
        $order_sum = $sum_with_sale;
        $sale = $sum - $sum_with_sale;

    } else {
        $order_sum = $sum;
        $sale = 0;
    }

    //Сохраняем информацию в заказы
    $order_info = array('sum' => $order_sum,
        'sale' => $sale,
        'date' => $date,
        'time' => $time

```

```

        );

        $order = ORM::factory('Order');
        $order->values($order_info)->save();

        Controller::redirect('shop/menu');

    } else {
        Controller::redirect('shop/menu');
    }

} //End action_order

```

*Продолжение приложения Б*

```

public function action_active_orders() {

    $this->template->title = 'АКТИВНЫЕ заказы';

    //Дата и время
    $today = date("Y-m-d");

    $order = ORM::factory('Order');

    $last_orders_1 = $order->get_current_orders($today, 10, 0);
    $last_orders_2 = $order->get_current_orders($today, 10, 10);

    $active_orders_1 = $last_orders_1;
    $active_orders_2 = $last_orders_2;
    $current_order_number = ORM::factory('Order')->get_this_number();
    $content = View::factory('shop/active_orders')
        ->bind('active_orders_1', $active_orders_1)
        ->bind('active_orders_2', $active_orders_2)
        ->bind('current_order_number', $current_order_number);
    $this->template->content = $content;
    $this->template->styles = array('/files/shop/css/active_orders.css' =>
'screen');
    $this->template->scripts = array('/files/shop/js/jquery-1.8.js',
        '/files/shop/js/active_orders.js');

} //End action_active_orders

```



```

} //End Shop
<?php defined('SYSPATH') or die('No direct script access.');
```

class Controller\_Shop\_Shoptemplate extends Controller\_Template {

```

    public $template = 'shop/shoptemplate';

    public function before() {

        parent::before();

        if (Auth::instance()->logged_in('login')) {
            Продолжение приложения Б

            $this->template->title = "";
            $this->template->content = "";
            $this->template->styles = array();
            $this->template->scripts = array();

            $styles = array();

            $scripts = array();

            $this->template->styles = array_merge( $this->template->styles, $styles );
            $this->template->scripts = array_merge( $this->template->scripts,
$scripts );

        } else {
            Controller::redirect('login');
        }

    } //End Function before

} //End Template_Controller Shoptemplate
<?php defined('SYSPATH') or die('No direct script access.');
```

/\*  
\* Контроллер обрабатывающий запросы от Ajax для редактирования  
\* данных в Админ-панели  
\*/

```

class Controller_AjaxAdmin extends Controller {
```

```

public function before() {
    parent::before();

    if(Auth::instance()->logged_in('admin')) {
        //
    } else {
        die;
    }
}

public function action_save_price() {

    $data = $this->request->post();
        Продолжение приложения Б

    if(isset($data['id'])){
        $dish = ORM::factory('Dish')
            ->where('id', '=', $data['id'])
            ->find();
        $dish->set('cost', $data['price'])
            ->set('discount', $data['discount'])
            ->update();
        echo 1;
    }
} //End Action save_price

public function action_delete_product() {

    $data = $this->request->post();

    if(isset($data['id'])){
        ORM::factory('Dish')
            ->where('id', '=', $data['id'])
            ->find()
            ->delete();
        echo 1;
    }
} //End Action delete_product

/*
* Получение данных от Ajax и добавление нового продукта
* P.S. не без костылей :-D
*/
public function action_add_product() {

```

```

$data = $this->request->post();

if(isset($data['name']) && isset($data['cost']) && isset($data['discount'])) {

    $engname = $this->generate_name();

    ORM::factory('Dish')
        ->set('name', $data['name'])
        ->set('cost', $data['cost'])
        ->set('discount', $data['discount'])
        ->set('type', $data['type'])
        ->set('eng_name', $engname)
        Продолжение приложения Б
->set('image', $data['image'])
    ->create();

    echo 1;
}
} //End Action add_product

/*
 * ИЗМЕНЕНИЕ СТАТУСА ЗАКАЗА
 */
public function action_order_status() {

    $data = $this->request->post();

    if(isset($data['id'])){

        $order = ORM::factory('Order');
        $order ->where('id', '=', $data['id'])
            ->find()
            ->set('status', $data['status'])
            ->update();

        $stats = ORM::factory('Stat');
        $materials = ORM::factory('Material');

        //Добавляем или удаляем сводную статистику по продуктам и
        материалам
        if($data['status'] == 'complete') {
            $stats->add_stats($data['id']);
        }
    }
}

```

```

        $materials->add_material_stats($data['id']);
    } else {
        $stats->delete_stats($data['id']);
        $materials->delete_material_stats($data['id']);
    }
}

```

```

} //End Action change_status

```

```

/*
 * Генерация случайной строки
 * ЭПИК КОСТЫЛЬ
 */

```

*Продолжение приложения Б*

```

public function generate_name() {
    $arr = array('a', 'b', 'c', 'q', 'w', 'e', 'r', 't', 'y', 'u', 'i', 'o', 'p', 's', 'd', 'f', 'g', 'h', 'j', 'k',
    'l', 'z', 'x', 'v', 'n', 'm', '1', '2', '3', '4', '5', '6', '7', '8', '9', '0');
    $r = "";
    for ($i = 0; $i < 15; $i++) {
        $r.= $arr[rand(0, sizeof($arr) - 1)];
    }
    return $r;
}

```

```

} //End Controller AjaxAdmin
<?php defined('SYSPATH') or die('No direct script access.');
```

```

class Controller_Login extends Controller_Logintemplate {

```

```

    /*
    * Контроллер входа в приложение
    */
    public function action_index() {
        if (Auth::instance()->logged_in('login')) {
            Controller::redirect('shop');
        } elseif (Auth::instance()->logged_in('admin')) {
            Controller::redirect('admin/stats');
        } else {
            $this->template->title = 'Страница входа';
            $content = View::factory('login');
            $this->template->content = $content;
        }
    }
}

```

```

} //End Action index

/*
 * Проверка логина и пароля
 */
public function action_check() {

    $data = $this->request->post();

    if(isset($data['email']) || isset($data['password'])){
        $success = Auth::instance()->login($data['email'], $data['password']);

        $this->go_to($success);
            Продолжение приложения Б
    } else {
        Controller::redirect('login');
    }

} //End Action check

/*
 * Выход из приложения
 */
public function action_logout(){

    $logout = Auth::instance()->logout();
    if($logout) {
        Controller::redirect('login');
    }
} //End Action logout

/*
 * Переадресация после входа в приложение
 */
protected function go_to($success) {
    if ($success && Auth::instance()->logged_in('admin')) {
        Controller::redirect('admin/stats/summary');
    } elseif ($success && Auth::instance()->logged_in('login')) {
        Controller::redirect('shop');
    } else {
        Controller::redirect('login');
    }
}

```

```

    }//End Function go_to

} //End Login

<?php defined('SYSPATH') or die('No direct script access.');
```

class Controller\_Logintemplate extends Controller\_Template {

```

    public $template = 'logintemplate';

    public function before() {

        Продолжение приложения Б

parent::before();

        $this->template->title = "";
        $this->template->content = "";
        $this->template->styles = array();
        $this->template->scripts = array();

        $styles = array(
            '/files/login/css/demo.css' => 'screen',
            '/files/login/css/font-awesome.css' => 'screen',
            '/files/login/css/style.css' => 'screen',

        );

        $scripts = array(
            '/files/login/js/jquery.placeholder.min.js',
            '/files/login/js/modernizr.custom.63321.js',
        );

        $this->template->styles = array_merge( $this->template->styles, $styles );
        $this->template->scripts = array_merge( $this->template->scripts, $scripts );

    }

} //End Logintemplate

<?php defined('SYSPATH') or die('No direct script access.');
```

/\*

```
* Контроллер обрабатывающий запросы от Ajax для добавления,  
* удаления и очистки "временных" продуктов  
*/
```

```
class Controller_SendData extends Controller {
```

```
    public function before() {  
        parent::before();  
  
        if(Auth::instance()->logged_in('login')) {  
            //  
        } else {  
            die;
```

*Продолжение приложения Б*

```
    }  
}
```

```
/*  
* Добавление продукта в корзину  
*/
```

```
public function action_add_data() {
```

```
    $data = $this->request->post();
```

```
    if(isset($data['data'])){
```

```
        //Получаем запрос от Ajax и находим нужных продукт
```

```
        $eng_name = $data['data'];
```

```
        $dish = ORM::factory('Dish')
```

```
            ->where('eng_name', '=', $eng_name)
```

```
            ->find();
```

```
        $price = $dish->cost;
```

```
        $name = $dish->name;
```

```
        $discount = $dish->discount;
```

```
        $dish_id = $dish->id;
```

```
        $image = $dish->image . ".png";
```

```
        $type = $dish->type;
```

```
        //Вычисляем скидку и округляем до 5 в большую сторону:
```

```
        $price_discount = $price - ($price * $discount / 100);
```

```
        $modul = $price_discount % 5;
```

```
        if($modul !== 0) {
```

```
            $price_discount += 5 - $modul;
```

```

}

//Генерируем нужный order_id для продукта
$new_order_id = ORM::factory('Order')->get_order_id();

//Создаем массив данных и сохраняем его в продукты
$product_info = array('order_id' => $new_order_id,
    'dish_id' => $dish_id,
    'product' => $name,
    'type' => $type,
    'price' => $price,
    'discount' => $discount,
    Продолжение приложения Б
    'price_discount' => $price_discount
);

$product = ORM::factory('Product');

$product->values($product_info)
->save();

//Получаем сумму продуктов для текущего заказа
$sum = $product->get_products_sum($new_order_id);

//"отдаем" данные Ajax'y
$response = array(
    'price'=> $price,
    'name'=> $name,
    'sum' => $sum,
    'image' => $image,
);

echo json_encode($response);

}
} //End Action add_data

```

```
/*
```

```
* Удаление продукта из корзины
```



```

*/
public function action_delete_data() {

    $data = $this->request->post();
    if(isset($data['delete1'])) {

        //Получаем запрос от Ajax и находим нужных продукт
        $eng_name = $data['delete1'];
        $dish = ORM::factory('Dish')
            ->where('eng_name', '=', $eng_name)
            ->find();
        $name = $dish->name;
    }
}

```

*Продолжение приложения Б*

```

//Генерируем нужный order_id для продукта
$new_order_id = ORM::factory('Order')->get_order_id();

$product = ORM::factory('Product');

//Удаление продукта для текущего заказа
$product
    ->where('order_id', '=', $new_order_id)
    ->and_where('product', '=', $name)
    ->or_where('id', '=', $eng_name)
    ->find()->delete();

if ($product->loaded()) {
    $result = 1;
} else {
    $result = 0;
}

//Получаем сумму и сумму со скидкой продуктов для текущего заказа
$sum = $product->get_products_sum($new_order_id);
$sum_with_sale = $product->get_products_sale($new_order_id);

//"отдаем" данные Ajax'у
$response = array(
    'sum' => $sum,
    'sum_with_sale' => $sum_with_sale,
    'result' => $result,
);

```

```
        echo json_encode($response);
    }

} //End Action delete_data
```

```
/*
 * Очистка корзины
 */
public function action_clear_all() {
```

```
    $data = $this->request->post();
    if(isset($data['clearAll'])) {
```

*Продолжение приложения Б*

```
$new_order_id = ORM::factory('Order')->get_order_id();
```

```
    $product = ORM::factory('Product');
    $product->where('order_id', '=', $new_order_id)
        ->delete_all();
```

```
    echo 1;
```

```
    } else {
        echo 0;
    }
```

```
} //End Action clear_all
```

```
/*
 * Изменение статуса заказа
 */
public function action_change_status() {
```

```
    $data = $this->request->post();
```

```
    if(isset($data['id'])) {
```

```
        $order = ORM::factory('Order');
        $order ->where('id', '=', $data['id'])
            ->find()
            ->set('status', $data['status'])
            ->update();
```

```

    $stats = ORM::factory('Stat');
    $materials = ORM::factory('Material');

    //Добавляем или удаляем сводную статистику по продуктам и
    материалам
    if($data['status'] == 'complete') {
        $stats->add_stats($data['id']);
        $materials->add_material_stats($data['id']);
    } else {
        $stats->delete_stats($data['id']);
        $materials->delete_material_stats($data['id']);
    }

```

*Продолжение приложения Б*

```

}

```

```

} //End Action change_status

```

```

/*

```

```

* Изменение статуса всех последних заказов

```

```

public function action_changeAll_status() {

```

```

    $data = $this->request->post();

```

```

    if(isset($data['status_all'])) {
        $today = date("Y-m-d");
        $order = ORM::factory('Order');
        $order ->order_by('id', 'DESC')
            ->limit(20)
            ->where('date', '=', $today)
            ->set('status', $data['status_all'])
            ->update_all();
    }

```

```

} //End Action changeAll_status

```

```

*/

```

```

} //End Controller SendData

```

```

<?php defined('SYSPATH') OR die('No direct access allowed.');
```

```

return array

```

```

(
    'default' => array
    (
        'type'    => 'MySQL',
        'connection' => array(
            /**
             * The following options are available for MySQL:
             *
             * string  hostname  server hostname, or socket
             * string  database  database name
             * string  username  database username
             * string  password  database password
             * boolean persistent use persistent connections?
             */
            Продолжение приложения Б
        * array  variables  system variables as "key => value" pairs
            *
            * Ports and sockets may be appended to the hostname.
            */
            'hostname' => 'localhost',
            'database' => 'bappetitdb_5',
            'username' => 'sunsystem',
            'password' => 'qwerty',
            'persistent' => FALSE,
        ),
        'table_prefix' => "",
        'charset'    => 'utf8',
        'caching'    => FALSE,
    ),
    'alternate' => array(
        'type'    => 'PDO',
        'connection' => array(
            /**
             * The following options are available for PDO:
             *
             * string  dsn      Data Source Name
             * string  username  database username
             * string  password  database password
             * boolean persistent use persistent connections?
             */
            'dsn'      => 'mysql:host=localhost;dbname=kohana',
            'username' => 'root',
            'password' => 'r00tdb',
            'persistent' => FALSE,
        )
    )
)

```

```
),
/**
 * The following extra options are available for PDO:
 *
 * string identifier set the escaping identifier
 */
'table_prefix' => "",
'charset'      => 'utf8',
'caching'      => FALSE,
),
);
```