

МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ  
И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН

САМАРКАНДСКИЙ ФИЛИАЛ ТАШКЕНТСКОГО УНИВЕРСИТЕТА  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫЙ ИНЖИНИРИНГ

КАФЕДРА ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

ВЫПУСКНАЯ

КВАЛИФИКАЦИОННАЯ РАБОТА

для получения академической степени бакалавра по направлению  
5610600 - «Телекоммуникационные технологии и профессиональное  
образование»

**ТЕМА: ФОРМИРОВАНИЕ БАЗЫ ДАННЫХ АНАЛИЗА И ОЦЕНКИ  
ФУНКЦИОНИРОВАНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ  
БЫТОВЫХ УСЛУГ.**

Работа рекомендовано к  
защите заседанием кафедры  
№ \_\_\_\_ от \_\_\_\_ мая 2016 года

Заведующий кафедрой

\_\_\_\_ ст. преп. Якубжанова Д.К.

“ \_\_\_\_ ” \_\_\_\_\_ 2016 г.

Выполнил: \_\_\_\_\_

студент 409-группы

Рашидов Бобир

Научный руководитель:

\_\_\_\_\_ Ганиева Н.А.

САМАРКАНД – 2016

## СОДЕРЖАНИЕ

<b>Введение .....</b>	<b>3</b>
<b>Глава 1. Основные характеристики информационных систем .....</b>	<b>5</b>
1.1 Характерные особенности и свойства информационных систем.....	8
1.2 Технология создания и управления базой данных .....	12
1.3 Основные понятия системы оказания бытовых услуг.....	
<b>Глава 2. Средства программирования и алгоритмы применяемые для создания информационной системы оказания услуг.....</b>	<b>18</b>
2.1. Алгоритм работы базы данных информационной системы оказания бытовых услуг.....	18
2.2. Функциональная структура программной системы обработки базы данных .....	20
<b>Глава 3. Вычислительный эксперимент автоматизации информационной системы оказания бытовых услуг.....</b>	<b>24</b>
3.1. Основы разработки базы данных и интерфейса программы.....	24
3.2. Программа автоматизации управления системы оказания бытовых услуг.....	29
3.3. Техника безопасности при работе на компьютере.....	47
<b>Заключение.....</b>	<b>50</b>
<b>Литература.....</b>	<b>51</b>
<b>Приложение.....</b>	<b>52</b>

## Введение

**Актуальность работы.** Постановлением главы государства утверждена также Программа дальнейшего внедрения и развития информационно-коммуникационных технологий в Республике Узбекистан на 2012-2014 годы, в которой подробно и адресно по годам прописаны организационные меры, развитие инфраструктуры ИКТ, подготовка, переподготовка и повышение квалификации кадров, совершенствование правительственного портала, веб-сайтов государственных органов и интерактивных услуг, обеспечение информационной безопасности, применение современных средств ИКТ в деятельности коммерческих структур, популяризация ИКТ среди населения, в том числе в сельской местности [1].

С развитием общества всё больше возникала потребность в широком обмене информацией. Однако только после изобретения письменности и книгопечатания, позволившего массово собирать и распространять данные, создание телеграфной и телефонной связи, передающей сообщения в течение минуты, возникла система распространения информации. Человек стал использовать для обработки информации различные технические устройства и приспособления в виде арифмометров, логарифмических линеек, калькуляторов, электронных вычислительных машин, ознаменовавших переход к информационному обществу.

Работа человека с информацией при использовании компьютерной техники предъявляет обществу новые требования, в связи с этим развивается информатизация.

Информатизацией общества можно назвать организованный социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационных потребностей и реализации прав граждан, организаций, объединений, органов

государственной власти и т.д. на основе формирования и использования соответствующих информационных ресурсов.

Информатизация – это сведения, данные, с помощью которых передаются накопленный опыт и знания [2].

Среди них особое место занимают базы данных. Их использование позволяет сократить время, требуемое на поиск нужной информации, уменьшить непроизводительные затраты при их реализации, исключить возможность появления ошибок в подготовке различных видов документации, что дает для предприятия прямой экономический эффект.

Основная причина применения систем, использующих базу данных, является стремление собрать все обрабатываемые данные в единое целое и обеспечить к ним контролируемый доступ.

**Цель и задачи исследования.** Целью данной выпускной квалификационной работы является разработка информационной системы оказания бытовых услуг, которая направлена на обслуживание населения и регулирования рынка бытовых услуг с целью обеспечения полного и комплексного обслуживания потребителей, улучшения качества жизни населения.

При разработке системы управления и оказания услуг были учтены следующие критерии:

- значение услуги для жизни и здоровья клиентов;
- социальная значимость;
- финансирование;
- территория обслуживания.

Для достижения поставленной цели необходимо решение конкретных задач:

- исследовать предметную область системы оказания услуг;
- построить инфологическую модель;
- построить даталогическую модели данных;
- спроектировать базу данных системы.

**Структура и объем работы.** Выпускная квалификационная работа состоит из введения, трёх глав, заключения, списка использованной литературы и приложения.

# ГЛАВА 1. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ИНФОРМАЦИОННЫХ СИСТЕМ.

## *1.1 Характерные особенности и свойства информационных систем*

Информационная система управления – совокупность информации, экономико-математических методов и моделей, технических, программных, других технологических средств и специалистов, предназначенная для обработки информации и принятия управленческих решений. Информационная система управления должна решать текущие задачи стратегического и тактического планирования, бухгалтерского учета и оперативного управления фирмой. Многие учетные задачи (бухгалтерского и материального учета, налогового планирования, контроля и т.д.) решаются без дополнительных затрат путем вторичной обработки данных оперативного управления. Учет является необходимым дополнительным средством контроля. Используя оперативную информацию, полученную в ходе функционирования автоматизированной информационной системы, руководитель может спланировать и сбалансировать ресурсы фирмы (материальные, финансовые и кадровые), просчитать и оценить результаты управленческих решений, наладить оперативное управление себестоимостью продукции (товаров, услуг), ходом выполнения плана, использованием ресурсов и т.д. Информационные системы управления позволяют:

- повышать степень обоснованности принимаемых решений за счет оперативного сбора, передачи и обработки информации;
- обеспечивать своевременность принятия решений по управлению организацией в условиях рыночной экономики;
- добиваться роста эффективности управления за счет своевременного представления необходимой информации руководителям всех уровней управления из единого информационного фонда;

–согласовывать решения, принимаемые на различных уровнях управления и в разных структурных подразделениях;

– за счет информированности управленческого персонала о текущем состоянии экономического объекта обеспечивать рост производительности труда, сокращение непроизводственных потерь и т.д.[2,14].

Системы управления заказами, системы управления взаимодействия с клиентами (CRM-системы) - корпоративные информационные системы, предназначенные для автоматизации CRM-стратегии компании, в частности, для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путём сохранения информации о клиентах (контрагентах) и истории взаимоотношений с ними, установления и улучшения бизнес-процедур и последующего анализа результатов.

Поддержка этих бизнес-целей включает сбор, хранение и анализ информации о потребителях, поставщиках, партнёрах, а также о внутренних процессах компании. Функции для поддержки этих бизнес-целей включают продажи, маркетинг, поддержку потребителей, управление качеством, обучение и повышение квалификации сотрудников компании, найм и развитие персонала, управление мотивацией персонала. Технологии для поддержки модели CRM должны являться частью общей клиентоориентированной стратегии компании.

Рассмотренные решения требуют больших временных и ресурсных затрат на внедрение и имеют высокую стоимость. Кроме всего, приведенный выше обзор аналогичных продуктов других производителей показал, что:

- некоторые приложения не имеют достаточно гибкой настройки;
- ограниченные функциональные возможности;
- не все бесплатные пакеты имеют полную документацию на русском языке, необходимую для разработки собственных модулей или изменения существующей программы.

Анализ вышеперечисленных недостатков аналогичных продуктов, а также требований конечных пользователей такого рода продуктов, позволяет сформулировать следующий список общих технических требований к разрабатываемой базе данных:

- ведение учета номерного фонда системы;
- ведения прейскуранта цен на услуги;
- ведение прейскуранта цен на дополнительные услуги;
- учет заявок;
- получение отчетности по выручке за период.

Проектируемая система должна быть масштабируемой и гибкой. Под масштабируемостью понимается возможность использовать ее на многократно возросших объемах операций бизнеса, а под гибкостью - наращивание функциональности системы и возможность ведения собственной разработки[2,3].

Процесс разработки данной информационной системы можно разделить на две части. Первая из них является проектной, вторая же направлена непосредственно на разработку соответствующего программного продукта.

Этап проектирования системы включает в себя решение следующих вопросов:

- исследование предметной области;
- обоснование необходимости разработки системы;
- постановка задачи и цели разработки;
- обзор научной литературы и анализ теоретического материала;
- анализ существующих систем-аналогов;
- проектирование структуры информационной системы;
- разработка методов и алгоритмов решения поставленной задачи.

Разработка программного комплекса включает в себя решение следующих задач:

- выбор языка программирования, информационной модели, комплекса технических средств;
- разработка структуры базы данных;
- определение стандартных запросов к системе, способов получения информации, форм предоставления информации;
- построение структуры программного комплекса, определение функций составных частей и их взаимосвязи;
- разработка, тестирование и отладка приложения и базы данных;
- создание документации на разработанную систему (руководства по эксплуатации, модификации, доработке).

Для реализации системы в соответствии с требованиями задания, необходимо разработать базу данных для хранения информации.

Хранение информации в базе данных имеет следующие преимущества перед другими способами организации хранения:

- легко организовывать поиск, сортировать записи по дате и времени, проводить различные отборы записей, так как в БД предусмотрена эффективная организация хранения информации, минимизирующая время доступа и поиска;
- запись в БД могут осуществлять неограниченное число пользователей. В файлы же, если идёт запись одним, то другой пользователь получит отказ;
- возможность устранения противоречивости;
- введение ограничений для обеспечения целостности данных;

### ***1.2 Технология создания и управления базой данных.***

Современные формы информационных систем являются банки данных, включающие в свой состав следующие составляющие: вычислительную систему, систему управления базой данных, одну или несколько баз данных, набор прикладных программ.

База данных – это совместно – используемый набор логически связанных данных и описание этих данных, которые предназначены для удовлетворения информационных потребностей организации фирмы.

База данных – это единое, большое хранилище данных, которое однократно определяется, а затем используется одновременно многими пользователями из разных подразделений. Вместо разрозненных файлов с избыточными данными, здесь все данные собраны вместе с минимальной долей избыточности. База данных хранит не только рабочие данные организации, но и их описания. Информация в базе данных должна отвечать многим требованиям, основными из них являются: неизбыточность, непротиворечивость, целостность[8].

Система управления базами данных (СУБД) – это программное обеспечение, с помощью которого пользователи могут определять, создавать и поддерживать базу данных, а также осуществлять к ней контролируемый доступ.

СУБД – это программное обеспечение, которое взаимодействует с прикладными программами пользователей и базой данных и обладает следующими возможностями:

- позволяет определять базу данных;
- позволяет вставлять, обновлять, удалять и извлекать информацию из базы данных;
- предоставляет контролируемый доступ к базе данных.

Обладание такими функциональными возможностями превращает СУБД в чрезвычайный полезный инструмент.

Преимущества СУБД:

- контроль за избыточностью данных;
- непротиворечивость данных;
- больше полезной информации при том же объеме хранимых данных;
- совместное использование данных;

- поддержка целостности данных;
- повышенная безопасность;
- применение стандартов;
- повышение эффективности с ростом масштабов системы;
- возможность нахождения компромисса при противоречивости в требованиях;
- повышение доступности данных и готовности к работе;
- улучшение показателей производительности;
- упрощение сопровождения системы за счет независимости от данных;
- улучшенное управление параллельностью;
- развитые службы резервного копирования и восстановления.

#### Недостатки СУБД:

- сложность;
- размер;
- стоимость СУБД;
- дополнительные затраты на аппаратное обеспечение;
- затраты на преобразование;
- производительность;
- более серьезные последствия при выходе системы из строя.

*Виды баз данных. Виды моделей данных. Виды баз данных.*

Классификация баз и банков данных может быть произведена по разным признакам, относящимся к разным компонентам и сторонам функционирования банков данных, среди которых выделяют, например, следующие.

По форме представляемой информации можно выделить фактографические, документальные, мультимедийные, в той или иной степени соответствующие цифровой, символьной и другим (нецифровой и не символьной) формам представления информации в вычислительной среде. К

последним можно отнести картографические, видео-, аудио-, графические и другие БД.

По типу хранимой (не мультимедийной) информации можно выделить фактографические, документальные, лексикографические БД. Лексикографические базы — это классификаторы, кодификаторы, словари основ слов, тезаурусы, рубрикаторы и т. д., которые обычно используются в качестве справочных совместно с документальными или фактографическими БД. Документальные базы подразделяются по уровню представления информации на полнотекстовые (так называемые «первичные» документы) и библиографическо - реферативные («вторичные» документы, отражающие на адресном и содержательном уровнях первичный документ)[9].

По типу используемой модели данных выделяют три классических класса БД: иерархические, сетевые, реляционные. Развитие технологий обработки данных привело к появлению постреляционных, объектно-ориентированных, многомерных БД, которые в той или иной степени соответствуют трем упомянутым классическим моделям.

По топологии хранения данных различают локальные и распределенные БД. По типологии доступа и характеру использования хранимой информации БД могут быть разделены на специализированные и интегрированные.

По функциональному назначению (характеру решаемых с помощью БД задач и, соответственно, характеру использования данных) можно выделить операционные и справочно-информационные. К последним можно отнести ретроспективные БД (электронные каталоги библиотек, БД статистической информации и т. д.), которые используются для информационной поддержки основной деятельности и не предполагают внесения изменений в уже существующие записи, например, по результатам этой деятельности. Операционные БД предназначены для управления различными технологическими процессами. В этом случае данные не только извлекаются из БД,

но и изменяются (добавляются) в том числе в результате этого использования.

По сфере возможного применения можно различать универсальные и специализированные (или проблемно-ориентированные) системы.

По степени доступности можно выделить общедоступные и БД с ограниченным доступом пользователей. В последнем случае говорят об управляемом доступе, индивидуально определяющем не только набор доступных данных, но и характер операций, которые доступны пользователю.

Следует отметить, что представленная классификация не является полной и исчерпывающей. Она в большей степени отражает исторически сложившееся состояние дел в сфере деятельности, связанной с разработкой и применением баз данных.

В последнем случае правильнее говорить об интегрированных информационных системах, объединяющих в общей среде разнородные данные, хранимые возможно в разнотипных базах, но используемых для решения одной прикладной задачи.

Модели данных. Модель данных – интегрированный набор понятий для описания данных, связей между ними и ограничений, накладываемых на данные в некоторой организации.

Модель является представлением «реального мира» объектов и событий, а также существующих между ними связей. Модель должна отражать основные концепции, представленные в таком виде, который позволит проектировщикам и пользователям базы данных обмениваться конкретными и недвусмысленными мнениями об их понимании роли тех или иных данных в этой организации. Цель построения модели данных заключается в представлении данных в понятном виде. Если такое представление возможно, то модель данных легко можно будет применить при проектировании базы данных.

### *Виды моделей данных.*

Объективные модели данных. При построении объектных моделей данных используются такие понятия как сущности, атрибуты и связи. Сущность – отдельный элемент организации, который должен быть представлен в базе данных. Атрибут – свойство, которое описывает некоторый аспект объекта и значение которого следует зафиксировать, а связь является ассоциативным отношением между сущностями. Общие типы объектных моделей данных: модель типа «сущность - связь», семантическая модель, функциональная модель и объектно-ориентированная модель.

Модели данных на основе записей. В модели на основе записи база данных состоит из нескольких записей фиксированного формата, которые могут иметь разные типы. Каждый тип записи определяет фиксированное количество полей, каждое из которых имеет фиксированную длину. Существует три типа логических моделей данных на основе записей: реляционная модель данных, сетевая модель данных и иерархическая модель данных.

Реляционная модель данных. Реляционная модель данных основана на понятии математических отношений. В реляционной модели данные и связи представлены в виде таблиц, каждая из которых имеет несколько столбцов с уникальными именами. В реляционной модели данных единственное требование состоит в том, что база данных с точки зрения пользователя выглядит как набор таблиц.

Сетевая модель данных. В сетевой модели данные представлены в виде коллекций записей, а связи – в виде наборов. Здесь явным образом моделируются наборы, которые реализуются с помощью указателей. Сетевую модель можно представить как граф с записями в виде узлов графа и наборами в виде его ребер.

Иерархическая модель данных. Иерархическая модель является ограниченным подтипом сетевой модели. В ней данные также представлены

как коллекции записей, а связи – как наборы. Однако в иерархической модели узел может иметь только одного родителя. Иерархическая модель может быть представлена как древовидный граф с записями в виде узлов и множествами в виде ребер.

### ***1.3 Основные понятия системы оказания бытовых услуг***

В середине двадцатого века отрасли, оказывающие услуги, постепенно превратились в развитых странах в доминирующий сектор национального хозяйства по производству валового внутреннего продукта и численности занятых, что получило название "сервисной революции". Сектор услуг в экономике западных стран переживает в настоящее время перерождение, в ходе которого на смену традиционным методам ведения бизнеса приходят новые. Внедрение в процессы обслуживания все более и более совершенных технологий и тщательно разработанных систем сервиса, отвечающих запросам взыскательных потребителей, служат пропуском на новые рынки услуг.

Развитие услуг в ближайшие годы "сервисная революция" ярко проявляется и у нас. Поэтому крайне необходимо изучить тенденции, сформулировать принципы и задачи по регулированию этого процесса. Прежде чем проводить такой анализ, необходимо определиться с терминологией, относящейся к объекту исследования.

Услуга, по общеупотребительному определению К. Маркса, это полезное действие той или иной потребительной стоимости, она выступает, с одной стороны, как действие, с другой стороны, как результат трудовой деятельности в виде полезного эффекта труда, который может воплощаться как в товарах, так и в нематериальной форме. Совокупность всех видов деятельности по оказанию услуг называют сферой услуг. Часть сферы услуг, деятельность которой направлена на обслуживание населения, принято называть сферой сервиса[5,6].

Раскрывая содержание сферы сервиса, многие исследователи отмечают в качестве первоочередной ее задачи оказание бытовых услуг населению. Их особенностью является направленность на обслуживание повседневных или периодически возникающих хозяйственно-бытовых потребностей человека.

Всю область жизнедеятельности человека можно обозначить категорией "образ жизни", охватывающей совокупность типичных видов деятельности индивида, социальной группы, общества в целом в единстве с условиями жизни. Применение этого понятия позволяет рассматривать во взаимосвязи основные стороны жизни людей: труд, быт, культуру, политическую жизнь. Каждая из этих составляющих помимо всего прочего формирует все разнообразие потребностей, возникающих в процессе жизнедеятельности. Среди сторон жизнедеятельности, включенных в понятие "образ жизни", отметим важную составляющую - быт. Быт складывается из множества разнообразных явлений, затрагивающих различные стороны жизни, потребности, возникающие в быту, во многих случаях носят весьма сложный и индивидуальный характер, а потому могут удовлетворяться только через специфический вид услуг.

Бытовые услуги – труд, воплощающийся в той или иной потребительной стоимости, которая способна удовлетворять следующие личные потребности:

- в товарах народного потребления, имеющих сугубо индивидуальное назначение, либо представляющих собой ранее созданные продукты, улучшенные или отремонтированные для дальнейшего потребления;
- в непроизводственной деятельности, которую семьи предпочитают в индивидуальном потреблении больше, чем результаты собственного труда;
- в некоторых дополнительных экономических результатах

производственного или непроизводственного труда вследствие появления новых потребностей или недостаточности имеющихся услуг для удовлетворения уже существующих потребностей человека.

## **ГЛАВА 2. СРЕДСТВА ПРОГРАММИРОВАНИЯ И АЛГОРТМЫ ПРИМЕНЯЕМЫЕ ДЛЯ СОЗДАНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОКАЗАНИЯ УСЛУГ**

### ***2.1. Алгоритм работы базы данных информационной системы оказания бытовых услуг.***

За последние годы сильно изменились технологии обслуживания, появились принципиально новые виды услуг, что требует кроме учета факторов периодичности использования услуг еще и учета места ее исполнения. В связи с этим разработана новая классификация услуг по размерам территории обслуживания:

1. Услуги, выполняемые на дому у клиента приходящим мастером (ремонт крупной бытовой техники, уборка дома и т.п.). Территория обслуживания определяется соотношением цены услуг и затратами мастера на дорогу.
2. Услуги каждодневного и повседневного спроса, выполняемые на предприятии услуг (мелкий и срочный ремонт обуви, услуги парикмахерских и т.п.). Они должны быть максимально приближены к потребителям, т.е. находиться в пределах пешеходной досягаемости 15-20 минут.
3. Услуги периодического спроса (пошив одежды, услуги химчисток и т.п.) имеют большой радиус действия, который составляет обычно 1-2 района, и размещаются на оживленных проспектах, площадях, улицах, вблизи остановок.
4. Услуги эпизодического спроса (крупный ремонт бытовой техники, центральная химчистка и др.), оказываемые предприятиями районного, городского и областного радиуса обслуживания следует размещать в торговых и культурных центрах района, городов, на главных магистралях, совмещая с другими предприятиями сферы обслуживания.
5. Услуги, для оказания которых требуется перемещение клиента к месту оказания услуги, т.к. используются ресурсы территории (туристско-

экскурсионные, оздоровительные и т.п.). Территория, на которой они оказываются и определяет размеры зоны обслуживания.

6. Услуги, для которых личная встреча клиента с мастером необязательна, т.к. используются современные системы связи – почта, телефон-факс, интернет, удаленные терминалы (банковские, информационно-консультационные, учебные, рекламные и т.п.) имеют практически неограниченную территорию обслуживания.

Рассматривая взаимосвязанные показатели " метод управления – уровень управления" управляемость основных видов услуг с учетом вышеизложенных предпосылок может быть отражена в виде классификации "услуги – уровень управления – метод управления".

Проектирование структуры базы данных выполнено на основании анализа предметной области, произведенного выше.

Категория "Вид услуг" имеет следующие атрибуты:

- наименование вида услуги;
- текущая стоимость услуги.

Категория "Клиент" может характеризовать как клиентов-физических лиц, так и юридических лиц. Данная категория имеет следующие атрибуты:

- наименование юр. лица или ФИО физического лица;
- серия паспорта (для физических лиц);
- номер паспорта (для физических лиц);
- ИНН (для юридических и физических лиц);
- расчетный счет (для юридических лиц).

Требование к функциональным характеристикам – программа «Информационная система оказания бытовых услуг» должна выполнять следующие функции[6,7]:

– данные, вносимые пользователем должны автоматически заноситься в базу данных;

- необходим контроль правильности вносимых данных, в случае ошибки, данные не должны заноситься в базу данных;
- должна быть возможность добавлять в базу данных новые записи, удалять устаревшие или не нужные записи, редактировать существующие;
- выполнять необходимые запросы на получение интересующих данных;
- программа должна иметь удобный и достаточно простой интерфейс, который будет понятен и не квалифицированному пользователю.
- получение перечня и общего числа специалистов фирмы, оказывающих услуги;
- получение списка занятых сейчас специалистов, которые освобождаются к данному сроку;
- получение данных об оказанных услугах данной фирмой за указанный период;
- получение списка недовольных клиентов и их жалобы;
- получение сведений о клиентах, с которыми заключены договора;
- получение сведений о новых клиентах за указанный период;
- получение сведений о конкретном человеке, сколько раз он подавал заявку, в какой период, какие оплачивал услуги;
- получение сведений о конкретном виде услуг, кем он был сделан и в какой период;

Выходные данные организуются в виде ответа на запрос, который выводится на главное окно.

## ***2.2. Функциональная структура программной системы обработки базы данных.***

База данных должна соответствовать требованиям заказчика и быть адекватной той предметной области, для которой она предназначена. Также выбранная СУБД должна легко подстраиваться под постоянно изменяющиеся процессы в информационной системе. Другими словами,

СУБД должна удовлетворять текущим требованиям пользователей и вновь возникающим, связанным с развитием фирмы, т.е. должна быть масштабируема (соответствовать растущей ИС) без глобальных перестроек и модификаций.

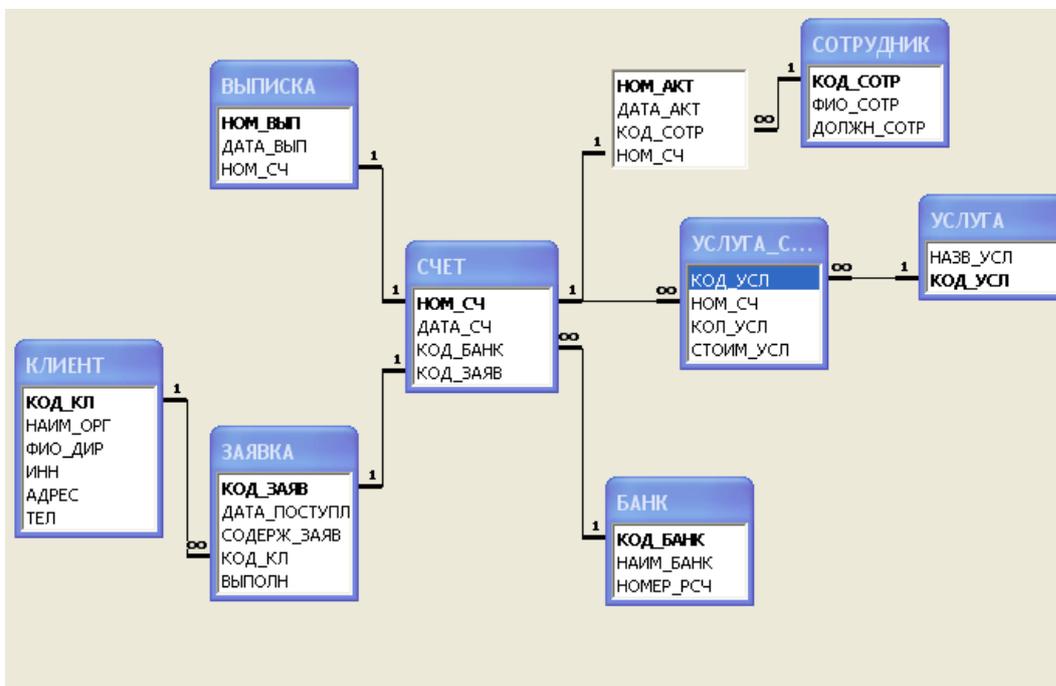


Рис.1 Структура базы данных системы управления.

**Учет операций, производимых с клиентами.** В первую очередь от клиента поступает заявка на обслуживание руководителю отдела сопровождения. При поступлении заявки руководитель отдела сверяет ее с заключенным договором на обслуживание, где выявляет, по каким услугам данный клиент может обслуживаться, по каким - не может. После этого руководитель отдела назначает работника на выполнение данной заявки[8,15].

Наим	Тип	Знач	Характеристики атрибутов
------	-----	------	--------------------------

Имя атрибута	Тип данных	Значение по умолчанию	Обязательность	Уникальность (Да/Нет)	Простой/Составной	Мнозначный/Однозначный	Статический/Динамический	Вычисляемый/Исходный	Условный (Да/Нет)
Код услуги	числовой	-	NOT NULL	Да	П	О	С	И	Нет
Название услуги	текстовый	-	NOT NULL	Да	П	О	С	И	Нет

Рис.2. Документирование атрибутов категории «Услуга»

После выполнения заявки руководитель отдела отмечает, что заявка выполнена. Работник отдела обращается в бухгалтерию с информацией о выполненной работе. Бухгалтерия выписывается два экземпляра счета, которые затем подписывает директор, и два экземпляра акта. Один экземпляр счета отправляется в архив счетов. Другой экземпляр счета и два экземпляра акта отдаются работнику, который обязан передать эти документы клиенту, перед этим подписав акты. После подписания клиент обязан возвратить акт выполненных работ в организацию. Оплата услуг производится через банк.

Имя атрибута	Тип данных	Значение по умолчанию	Характеристики атрибутов						
			Обязательность	Уникальность (Да/Нет)	Простой/Составной	Мнозначный/Однозначный	Статический/Динамический	Вычисляемый/Исходный	Условный (Да/Нет)
Номер счета	текстовый	-	NOT NULL	Да	П	О	С	И	Нет
Дата	дата	-	NOT	Нет	П	О	С	И	Нет

счет а			NUL L						
Код заявк и	число вой	-	NOT NUL L	Нет	П	О	С	И	Нет
Код банк а	число вой	-	NOT NUL L	Нет	П	О	С	И	Нет
Коли чест во услу ги	число вой	-	NOT NUL L	Нет	П	О	С	И	Нет
Стои мост ь услу ги	дене жный	-	NOT NUL L	Нет	П	О	С	И	Нет

Рис.3 Документирование атрибутов категории «Счет».

Также производится контроль оплаты клиентом услуг. Руководитель отдела по запросу получает из бухгалтерии выписку об оплате, которую в свое время бухгалтерия получила от банка, и отвечает в соответствующей заявке, что она выполнена и оплачена.

В БД должна храниться информация, достаточная для удовлетворения информационных потребностей в части отчетов и запросов.

## ГЛАВА 3. ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ АВТОМАТИЗАЦИИ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОКАЗАНИЯ БЫТОВЫХ УСЛУГ

### *3.1. Основы разработки базы данных и интерфейса программы.*

Для того, чтобы инструменты были выдержаны в стандартном стиле таких редакторов, работающих в среде Windows, как MS Word, а также имели поддержку со стороны операционной системы, при их создании использовались OLE-интерфейсы системных средств MSDHTML и MSIE. Исходя из перечня предоставляемых в распоряжение разработчика ресурсов этих средств были созданы классы, позволяющие отображать на экране и редактировать документы HTML на естественном языке пользователя[10,12].

**Основы HTML.** HTML - язык гипертекстовой разметки. Язык разметки исходного текста веб-документа, включающий специальные символы (теги), которые позволяют веб-браузеру сконструировать из текста дизайн. HTML предоставляет возможности форматирования и обработки форм, управления шрифтами, отображения информации в табличном виде, гипертекстовые связи и поддержку Java-апплетов.

Каждый HTML-документ, отвечающий спецификации HTML какой-либо версии, обязан начинаться со строки декларации версии HTML `<!DOCTYPE>`, которая обычно выглядит так: `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`

Эта строка поможет браузеру определить, как правильно интерпретировать данный документ. В данном случае мы говорим браузеру, что HTML соответствует международной спецификации версии 4.1, которая хоть и не отличается новизной, но, в отличие от более поздних версий, является полноценным, широко распространенным стандартом без каких-либо неопределенностей.

После объявления версии и типа документа необходимо обозначить его начало и конец. Это делается с помощью тэга-контейнера <HTML>. Необходимо отметить, что любой HTML-документ открывается тэгом <HTML> и им же закрывается. Эти тэги сообщают браузеру, что текст между ними следует интерпритировать как HTML-текст. Поскольку документы HTML чисто текстовые, тэг говорит о том, что файл написан на языке HTML (HyperTextMarkupLanguage - Язык гипертекстовой разметки).

Язык сценариев JavaScript. Язык программирования JavaScript разработан фирмой Netscape для создания интерактивных HTML-документов. Это объектно-ориентированный язык разработки встраиваемых приложений, выполняющих как на стороне клиента, так и на стороне сервера. Многие люди считают, что JavaScript - это то же самое, что и Java, лишь потому, что эти языки имеют схожие названия. На самом деле это не так. Клиентские приложения выполняются браузером просмотра Web-документов на машине пользователя, серверные приложения выполняются на сервере [11,12].

Основные области использования языка JavaScript при создании интерактивных HTML-страниц:

Динамическое создание документа с помощью сценария

Оперативная проверка достоверности заполняемых пользователем полей форм HTML до передачи их на сервер

Создание динамических HTML-страниц совместно с каскадными таблицами стилей и объектной моделью документа.

Взаимодействие с пользователем при решении “локальных” задач, решаемых приложением JavaScript, встроенном в HTML-страницу

Как и любой другой язык программирования, JavaScript использует переменные для хранения данных определенного типа. Реализация JavaScript является примером языка свободного использования типов. В нем не

обязательно задавать тип переменной. Ее тип зависит от типа хранимых в ней данных, причем при изменении типа данных меняется и тип переменной.

JavaScript поддерживает четыре простых типа данных:

Целый

Вещественный

Строковый

Булевый, или логический

Код на JavaScript обычно состоит из двух частей:

Функции, которые вызываются из кода странички в ответ на какое-то событие;

Код событий, которые вызывают функции.

Основы JQuery

jQuery — библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API по работе с Ajax.

Точно так же, как CSS отделяет визуализацию от структуры HTML, JQuery отделяет поведение от структуры HTML. Например, вместо прямого указания на обработчик события нажатия кнопки, управление передается JQuery, идентифицирующей кнопки и затем преобразовывающий его в обработчик события клика. Такое разделение поведения и структуры также называется принципом ненавязчивого JavaScript.

Библиотека jQuery содержит функциональность, полезную для максимально широкого круга задач. Тем не менее, разработчиками библиотеки не ставилась задача совмещения в jQuery функций, которые подошли бы всюду, поскольку это привело бы к большому коду, большая часть которого не востребована. Поэтому была реализована архитектура компактного универсального ядра библиотеки и плагинов. Это позволяет

собрать для ресурса именно ту JavaScript-функциональность, которая на нём была бы востребована [11].

Типичный пример манипуляции сразу несколькими узлами DOM заключается в вызове \$ функции со строкой селектора CSS, что возвращает объект jQuery, содержащий некоторое количество элементов HTML-страницы. Эти элементы затем обрабатываются методами jQuery. Например,

`$("#div.test").add("p.quote").addClass("blue").slideDown("slow");` находит все элементы div с классом test, а также все элементы p с классом quote, и затем добавляет им всем класс blue и визуально плавно спускает вниз. Здесь методы add, addClass и slideDown возвращают ссылку на исходный объект \$("#div.test"), поэтому возможно вести такую цепочку.

### ***3.2. Программа автоматизации функционирования информационной системы бытовых услуг.***

При разработке программы «Функционирования информационной системы бытовых услуг» были учтены основные требования пользовательского интерфейса.

При загрузке программы открывается главное окно, через которое пользователь имеет возможность перейти на другие категории.

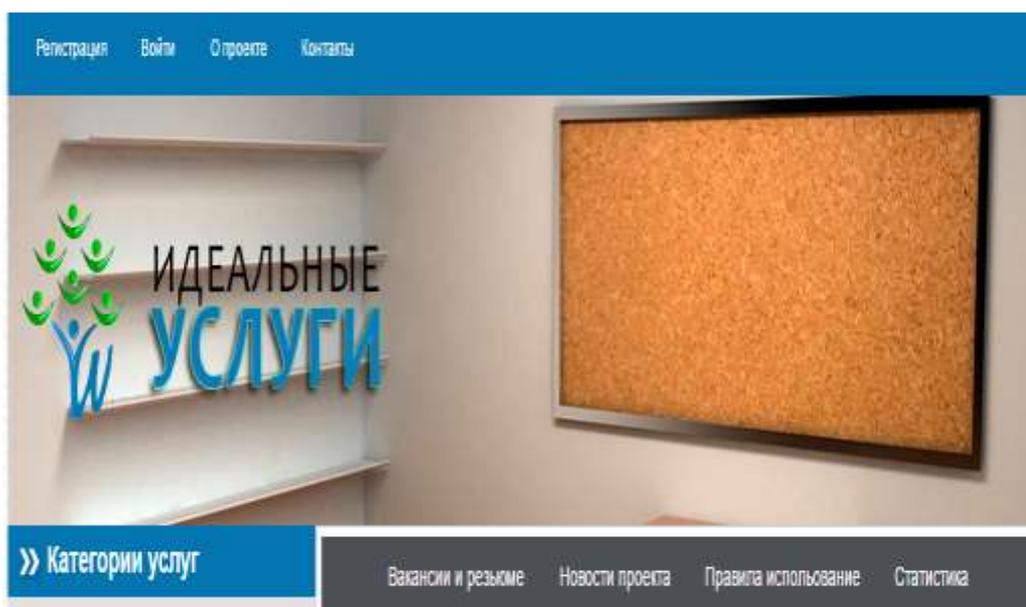


Рис.3.3.1. Главное окно программы.

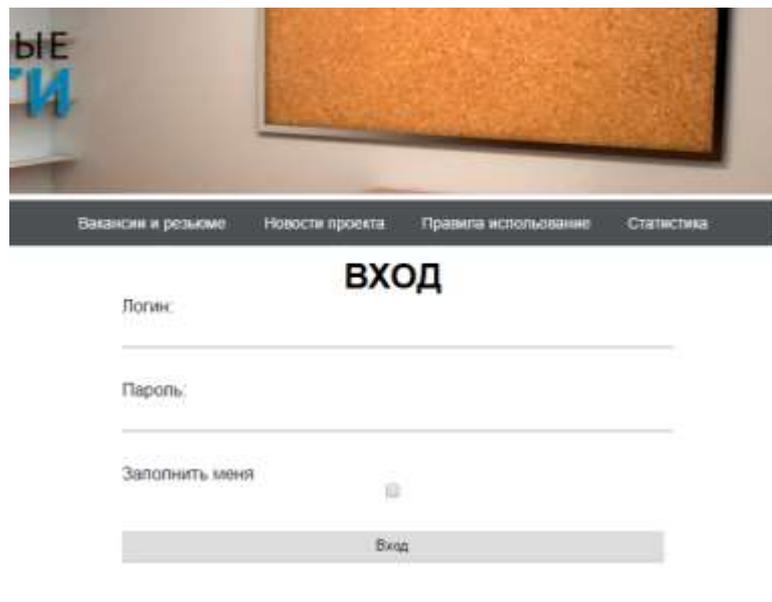


Рис. 3.3.2. Окно входа в программу системы управления.



Рис.3.3.3. Панель категорий услуг.

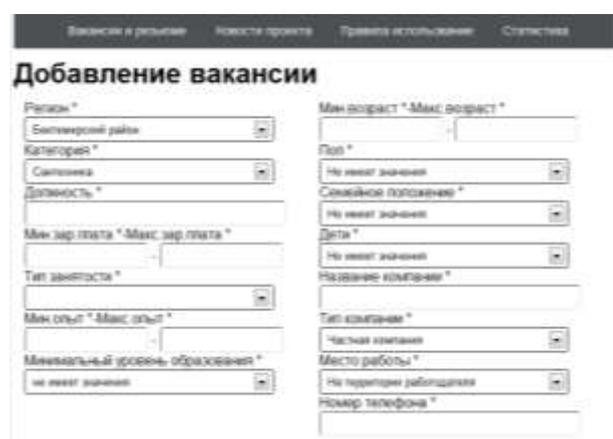


Рис.3.3.4. Окно бланка для предоставления услуг.

Ф.И.О.	Адрес	Контактный номер	Специализация	Цена услуги
Шарипов Сабир	Самарканд	2963344	Электрик, мастер по ремонту электрооборудований	400000
Компания "На долу"	Боттоварский район	8336(2345566)	Бухгалтер, финансист, прораб и т.д.	600000
Абдуллаев Шахриддин	Мира Уллубековский район	2345678	Сварщик	300000
Самаров Сабир	Самарканд	2333333	Сантехник	1500000

Рис.3.3.5. Окно с данными специалистов.

### Отправить заявку

Формы оплаты:

форма оплаты наличными образом Квитанция

форма оплаты с перечислениями Сканировать договор

форма оплаты посредством UZ CARD Оплата с системы

Ваше имя \*

Электронный адрес или телефон номер \*

День и время \*

Кратко о решаемой проблеме \*

Проверка кода  
 Получить новый код

Отправить

Рис.3.3.6. Бланк заявки.

### 3.3. Техника безопасности при работе на компьютере

Анализ опасных и вредных факторов, возникающих при работе на компьютер. Исследовательская работа в рамках данного проекта заключается в выполнении многих этапов, практически все из которых проходят в тесном контакте с ЭВМ. Длительная работа инженера-программиста с компьютером сопряжена с целым рядом вредных и опасных факторов. Рассмотрим некоторые из них [13].

Постоянное напряжение глаз. Работа с компьютером характеризуется высокой напряженностью зрительной работы. В выполняемом исследовании значительный объем информации на разных стадиях обработки представлен в графической форме с большим количеством мелких деталей, что дает

серьезную нагрузку на зрение. Постоянное напряжение глаз может привести к снижению остроты зрения. Экран видеомонитора должен находиться от глаз пользователя на оптимальном расстоянии 600...700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов. Также для снижения утомляемости рекомендуется делать 15-минутные перерывы в работе за компьютером в течение каждого часа [13].

Влияние электростатических и электромагнитных полей. Большинство ученых считают, что как кратковременное, так и длительное воздействие всех видов излучения от экрана монитора не опасно для здоровья персонала, обслуживающего компьютеры. Однако исчерпывающих данных относительно опасности воздействия излучения от мониторов на работающих с компьютерами не существует и исследования в этом направлении продолжаются.

Допустимые значения параметров неионизирующих электромагнитных излучений от монитора компьютера представлены в табл.1.

Максимальный уровень рентгеновского излучения на рабочем месте оператора обычно не превышает 10 мкбэр/ч, а интенсивность ультрафиолетового и инфракрасного излучений от экрана монитора лежит в пределах 10...100 мВт/м<sup>2</sup>.

Допустимые значения параметров неионизирующих электромагнитных излучений (в соответствии с СанПиН 2.2.2.542-96)

Таблица 1.

Наименование параметров	Допустимые значения
Напряженность электрической составляющей электромагнитного поля на расстоянии 50 см от поверхности видеомонитора	10 В/м

Напряженность магнитной составляющей электромагнитного поля на расстоянии 50 см от поверхности видеомонитора	0,3 А/м
Напряженность электростатического поля не должна превышать для взрослых пользователей для детей дошкольных учреждений, учащихся средних специальных и высших учебных заведений	20 кВ/м  15 кВ/м

Для снижения воздействия этих видов излучения рекомендуется применить мониторы с пониженным уровнем излучения (MPR-II, TCO-92, TCO-99), устанавливать защитные экраны, а также соблюдать регламентированные режимы труда и отдыха.

Длительное неизменное положение тела. Работа с компьютером характеризуется значительным умственным напряжением и нервно-эмоциональной нагрузкой операторов, высокой напряженностью зрительной работы и достаточно большой нагрузкой на мышцы рук при работе с клавиатурой ЭВМ. Большое значение имеет рациональная конструкция и расположение элементов рабочего места, что важно для поддержания оптимальной рабочей пользы человека-оператора.

Шум ухудшает условия труда, оказывая вредное действие на организм человека. Работающие в условиях длительного шумового воздействия испытывают раздражительность, головные боли, головокружение, снижение памяти, повышенную утомляемость, понижение аппетита, боли в ушах и так далее. В табл. 4.2 указаны предельные уровни звука в зависимости от категории тяжести и напряженности труда, являющиеся безопасными в отношении сохранения здоровья и работоспособности.

Техника безопасности при работе на ПК. Техника безопасности на ПК тесно связана с электробезопасностью. Большинство компьютеров имеют трехштырьковый разъем кабеля питания (в отличие от двухштырьковой вилки большинства бытовых приборов). Третий разъем - это так называемая «земля». В грамотно оборудованных компьютеризированных помещениях существует настоящая земля — заземляющий контур, который через металлическую ленту выводится на заземляющий штырь. Все это довольно громоздкое и дорогостоящее оборудование и часто разъем заземления не используется или используется «земляная» фаза обычной электросети. Результатом неправильного заземления могут быть удары статического напряжения от корпуса ПК. Частенько из-за этого сгорают сетевое оборудование компьютеров. Следует соблюдать не только правила электробезопасности, но и следить за состоянием здоровья обслуживающего персонала и пользователей ПК, защищая их от вредного воздействия электромагнитных излучений, заботясь о зрении, так как именно оно испытывает основные нагрузки при работе с вычислительной техникой. Важно также следить за состоянием техники и соблюдать основные правила обращения с ней.

Системный блок следует включать как можно реже (обычно включается в начале рабочего дня и выключается, выключается работа на нем — в конце дня). Для того, чтобы не выгорал экран и не расходовалась лишняя энергия, в компьютере предусмотрен специальный режим гашения экрана — через определенное время, если никто не работает на нем, т.е. нет обращения к клавиатуре или мыши, он выключается. Если монитор получает питание от системного блока, включая системный блок, включаем и монитор. Если соединение монитора и системного блока параллельно, то сначала необходимо включить монитор, потом системный блок. Выключать в обратной последовательности.

Экран монитора стеклянный, а потому и хрупкий, и поэтому надо обращаться с ним осторожно. Недопустимо попадание жидкости за заднюю часть экрана может замкнуть проводка, что выведет его из строя и может привести к возникновению пожара. В случае попадания жидкости следует отключить электропитание.

Защита от излучения расположена только на экране, поэтому, находясь прямо перед экраном, пользователь наиболее защищен от вредного воздействия излучения. На заднюю и боковые части монитора в целях экономии защиту не устанавливают. Следовательно, находясь сбоку или сзади монитора, можно получить максимально вредное воздействие.

При работе с клавиатурой стоит придерживаться следующих правил:

- 1) сильно не ударять по клавишам, это приводит к быстрой изнашиваемости прибора.
- 2) не распивать напитки над клавиатурой, так как попадание жидкости в нее приводит к короткому замыканию и выводит из строя клавиатуру, в случае попадания необходимо обесточить компьютер.
- 3) не кушать над клавиатурой бутерброды, семечки, так как крошки, попадающие в клавиатуру, нарушают ее работу.
- 4) при наличии защитной панели следует закрывать клавиатуру, тем самым, защищая ее от пыли.

Требования к параметрам микроклимата помещения. Под метеорологическими условиями (ГОСТ 12.1.005-88) понимают сочетание температуры, относительной влажности, скорости движения и запыленности воздуха. Перечисленные параметры оказывают огромное влияние на функциональную деятельность человека, его самочувствие и здоровье и на надежность средств вычислительной техники. Эти микроклиматические параметры влияют как каждый в отдельности, так и в различных сочетаниях[13].

Температура воздуха является одним из основных параметров, характеризующих тепловое состояние микроклимата. Суммарное тепловыделение в помещении поступает от:

- ЭВМ;
- вспомогательного оборудования;
- приборов освещения;
- людей;
- внешних источников.

Наибольшее количество теплоты выделяют ЭВМ и вспомогательное оборудование. Средняя величина тепловыделения от компьютеров колеблется до 100 Вт/м . Тепловыделения от приборов освещения также велики. Удельная величина их составляет 35 Вт/м . При этом, чем больше уровень освещенности, тем выше удельные величины тепловыделений. Количество теплоты от обслуживающего персонала незначительно. Оно зависит от числа работающих в помещении, интенсивности работы, выполняемой человеком.

К внешним источникам поступления теплоты относят теплоту, поступающую через окна от солнечной радиации, приток теплоты через непрозрачные ограждения конструкций. Интенсивность этих источников зависит от расположения здания, ориентации по частям света, цветовой гаммы и прочее [13].

Для обеспечения нормальных условий труда необходимо придерживаться вышеуказанных данных. В целях поддержания температуры и влажности воздуха в помещении можно использовать системы отопления, вентиляции и кондиционирования воздуха.

Параметры микроклимата производственных помещений

Параметры	Значения параметров	
	оптимальные	Допустимые

Температура	20-22 °С	17-22 °С
Относительная влажность	40-60 %	до 75%
Скорость движения воздуха	ОД м/с	не более 0,3 м/с

На исследуемом предприятии температура воздуха, влажность и скорость движения воздуха держится в рамках оптимальных параметров. Вредные вещества в воздухе рабочей зоны не превышают предельной допустимой концентрации [13].

## **Заключение**

Современный этап автоматизации управления общественным производством характерен развитием распределенной обработки экономической информации. Наиболее перспективной сферой использования концепции распределенной обработки экономической информации является автоматизация планово-управленческих функций на базе персональных компьютеров, установленных непосредственно на рабочих местах специалистов. Эти системы получили широкое распространение в организационном управлении под названием автоматизированных рабочих. Автоматизированное рабочее место представляет собой рабочее место персонала автоматизированной системы управления, оборудованное средствами, обеспечивающими участие человека в реализации функций управления.

В процессе работы над проектом проанализирована предметная область, выделены основные сущности, выбраны средства реализации базы данных и пользовательской программы. Произведено проектирование и реализация базы данных, а также реализация приложения для работы пользователей с базой данных.

Разработанная система позволяет собирать, хранить и анализировать информацию об оказанных услугах, вести журнал счетов, обеспечить трудоустройством. Она направлена на обслуживание населения и регулирования рынка бытовых услуг с целью обеспечения полного и комплексного обслуживания потребителей, улучшения качества жизни населения.

## Список литературы

1. Замоновий ахборот-коммуникация технологияларини янада жорий этиш ва ривожлантириш чора-тадбирлари тўғрисида. Ўзбекистон Республикаси Президентининг Қарори №ПҚ – 1730. Қабул қилинган сана 21.03.2012, кучга кириш санаси 21.03.2012. //Ўзбекистон Республикаси қонун ҳужжатлари тўплами, 2012 й., 13-сон, 139-модда.
2. Л.М. Чистов. Оптимизация управления социально-экономическими системами. Теория и практика. В 2 частях. - СПб: Изд-во СПбГУ, 1999.
3. Экономика бытового обслуживания. Под ред. В.Д Балалова. - М.: Легкая и пищевая промышленность, 1993.
4. Экономико-математическое моделирование региональных комплексов бытового обслуживания. В сб. Проблемы менеджмента и совершенствование маркетинговой деятельности в сфере услуг. М.: ГАСБУ, 1994.
5. Столмов Л.Ф. Изучение и прогнозирование спроса. - М., 1972.
6. Турков С.Л. Управление региональными программами экономического и социального развития: проблемы методологии и теории. Препринт. - Хабаровск: ВЦДВО РАН, 1998.
7. Казиев В.М. Введение в анализ, синтез и моделирование систем. Учебное пособие, ИНТУИТ.РУ, Бином. Лаборатория знаний, 2007
8. Голицина О. Л., Максимов Н. В. и др. База данных - Москва, 2003;
9. Коннолли Т. А., Бег К. В. Базы данных. Проектирование, реализация и сопровождение - Москва, 2001;
- 10/Благодатских В. А. и др. Стандартизация разработки программных средств - Москва, 2003;
11. Семакин И. Г., Шестаков А. П. Основы программирования - Москва, 2001
- 12 V.V.Merjevich. HTML Handbook, 2013.
13. Курс лекций «Техника безопасности при работе на компьютере».
14. <http://www.texnic.ru/books/opis/bgd/bgd001/bgd001.htm>
15. [www.allbest.ru](http://www.allbest.ru)

## ПРИЛОЖЕНИЕ

### Программы «Информационная система бытовых услуг»

```
* {  
  
    margin:0px;  
    padding:0px;  
    font-family: Book Antiqua;  
    font-size:12px/16px;  
    }  
  
.aa {  
    text-align:right;  
    }  
  
.aa a {  
    color:#000;  
    }  
  
@font-face {  
    font-family: 'Arab3end';  
    src: url('../fonts/arab3end.ttf') format('truetype');  
    font-weight: normal;  
    font-style: normal;  
    }  
  
body {  
background-image: url(../images/bg.png);  
    background-size: 100% auto;  
        background-attachment: fixed;  
        -moz-background-size: 100% auto;  
        -o-background-size: 100% auto;  
        -webkit-background-size: 100% auto;
```

```
-khtml-background-size: 100% auto;

}

.mainBody {
width:1024px;
margin:0px auto;
min-height: 700px;
box-shadow: 1px 1px 30px #fff;

position:relative;
z-index:1;
overflow:hidden;
}

.mainTitle {
font-size:3em;
}

.mainTitle a {
font-family:Calibri;
color:#00549D;
text-decoration:none;
position:absolute;
top:70px;
right:10px;
text-shadow: 1px 2px 8px #FFFFFF;
}

.abstract {
position:absolute;
z-index:5;
height:100%;
```

```

width:110px;
right:0px;
background:url(../images/abstract.png) repeat-y right top;
}
.topPart {
margin:0px auto;
width:1024px;
height: 267px;
position:relative;
overflow:hidden;
background: url(../images/banner.png) no-repeat;
}
.banner {
float:left;
width:650px;
height: 227px;
position:relative;
}
.bron {
margin-left:30px;
margin-top:10px;
float:left;
width:344px;
height: 230px;
box-shadow: 1px 3px 30px #333333;
}
.bronform {
margin-left:12px;
}

```

```
.formField {
    width: 210px;
    height:25px;
}

.formText {
    color:#FFF;
    font-size:13px;
    margin-top:3px;
    margin-bottom:5px;
}

.bronTitle {
    color:#FFF;
    margin-left:10px;
    margin-top:8px;
    margin-bottom:10px;
}

.formBtn {
    height:25px;
    margin-top:10px;
    background:url(../images/greybtn.png) no-repeat top;
    width: 210px;
    border-radius:30px;
    color:#FFF;
    font-size:16px;
}

.langBar {
    list-style:none;
    position:absolute;
    right:5px;
```

```

        top:5px;
    }
.langBar li {
    display:inline-block;
}
.langBar li a {
    color:#000;
    text-decoration:none;
    padding: 1px 3px;
    font-size: 16px;
}
.langBar li a:hover {
    text-decoration:underline;
}
.contacts {
    font-family:Arial, Helvetica, sans-serif;
    text-align:right;
    font-size:13px;
    color:#fff;
    text-shadow: 1px 2px 3px #000;
    position:absolute;
    right:5px;
    bottom:5px;
    line-height:15px;
}
.middlePart {
    width:1024px;
    min-height:700px;
    position:relative;

```

```

    }
.leftBlock {
    margin-top:-41px;
    margin-left:5px;
    float:left;
    width:650px;

    min-height:700px;
}
.rightBlock {
    margin-left:20px;
    float:left;
    width:343px;
    min-height:700px;
    color:#000;
}
.navigation {
    width:650px;
    height:30px;
    background:url(../images/navbg.png) repeat-x;
    margin-bottom:0px;
    border-top: 1px solid #84AED1;
    border-right: 1px solid #84AED1;
    border-left: 1px solid #84AED1;
}
.slider {
    border-bottom: 1px solid #84AED1;
    border-right: 1px solid #84AED1;
    border-left: 1px solid #84AED1;

```

```
        width:650px;
        height:250px;
    }
.contentPart {
    margin:0px auto;
    margin-top:-10px;
    width:620px;
    padding:15px;
}
.contentTitle {
    color:#003B93;
    font-size:18px;
    margin-top:15px;
}
.contentText {
    color:#000;
    font-size:14px;
    line-height:18px;
    text-indent: 40px;
    text-align:justify;
    margin-bottom:10px;
}
.line {
    border-top:3px solid #004297;
    width:95%;
    height: 5px;
    position:relative;
    margin:7px 0px;
}
```

```
.lineimg {
    position:absolute;
    right:-20px;
    top:-12px;
}

.rooms {
    width: 500px;
    font-size: 16px;
    color:#000;
}

.footer {
    margin-top: 20px;
    width:89%;
    height:50px;
    overflow:hidden;
    position:relative;
    border-top:6px solid #89A5BF;
}

.footerText {
    color:#FFF;
    font-size:14px;
    float:right;
    margin: 15px 30px 0px 0px;
}

.sidead {
    color:#000;
    margin-left:5px;
    margin-top:15px;
    float:left;
```

```
        position:relative;
        width:377px;
        height: 230px;
        box-shadow: 1px 3px 30px #333333;
        padding:15px 15px 15px 15px;
    }
    .imageRooms {
        width:220px;
    }
    .roomsTitle {
        color:#000;
        font-size:14px;
    }
    .roomsMore {
        position:absolute;
        bottom:15px;
        font-size:13px;
        right:160px;
    }
    .roomsMore a {
        text-decoration:none;
        color:#FFF;
    }
    .roomsMore a:hover {
        color:#FFF;
        text-decoration:underline;
    }
    .ngn {
        overflow:hidden;
```

```
    position:absolute;
    left:30px;
    height:50px;
    top:15px;
}
.ngn p a {
    text-decoration:none;
    color:#fff;}
.ngning {
    margin-bottom:-8px;
    margin-left:5px;}
```

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Menus, Grids, DBGrids, DB, DBTables, jpeg, ExtCtrls, ComCtrls,
StdCtrls;
```

```
type
```

```
TForm1 = class(TForm)
```

```
    MainMenu1: TMainMenu;
```

```
    N1: TMenuItem;
```

```
    N2: TMenuItem;
```

```
    N3: TMenuItem;
```

```
    N4: TMenuItem;
```

```
    N5: TMenuItem;
```

```
    N6: TMenuItem;
```

```
    N7: TMenuItem;
```

```
    N8: TMenuItem;
```

N21: TMenuItem;  
N9: TMenuItem;  
N22: TMenuItem;  
N10: TMenuItem;  
N11: TMenuItem;  
N12: TMenuItem;  
N23: TMenuItem;  
N31: TMenuItem;  
N41: TMenuItem;  
N51: TMenuItem;  
N61: TMenuItem;  
N71: TMenuItem;  
N81: TMenuItem;  
N91: TMenuItem;  
N101: TMenuItem;  
N111: TMenuItem;  
N121: TMenuItem;  
N131: TMenuItem;  
N141: TMenuItem;  
N151: TMenuItem;  
DBGrid1: TDBGrid;  
DataSource1: TDataSource;  
Query1: TQuery;  
StatusBar1: TStatusBar;  
Label1: TLabel;  
Label2: TLabel;  
N13: TMenuItem;  
Button1: TButton;  
N14: TMenuItem;

```
procedure N2Click(Sender: TObject);
procedure N3Click(Sender: TObject);
procedure N4Click(Sender: TObject);
procedure N6Click(Sender: TObject);
procedure N7Click(Sender: TObject);
procedure N8Click(Sender: TObject);
procedure N21Click(Sender: TObject);
procedure N9Click(Sender: TObject);
procedure N22Click(Sender: TObject);
procedure N10Click(Sender: TObject);
procedure N12Click(Sender: TObject);
procedure N31Click(Sender: TObject);
procedure N41Click(Sender: TObject);
procedure N61Click(Sender: TObject);
procedure N81Click(Sender: TObject);
procedure N71Click(Sender: TObject);
procedure N91Click(Sender: TObject);
procedure N51Click(Sender: TObject);
procedure N101Click(Sender: TObject);
procedure N121Click(Sender: TObject);
procedure N131Click(Sender: TObject);
procedure N141Click(Sender: TObject);
procedure N151Click(Sender: TObject);
procedure N23Click(Sender: TObject);
procedure N111Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure N14Click(Sender: TObject);
private
  { Private declarations }
```

```

public
  { Public declarations }
end;

var
  Form1: TForm1;

implementation

uses Unit11, Unit7, Unit3, Unit2, Unit4, Unit5, Unit6, Unit8, Unit9, Unit10,
  Unit12, Unit13, Unit14, Unit15, Unit16, Unit17, Unit18, Unit19, Unit21,
  Unit20, Unit22, Unit23, Unit24, Unit25, Unit26, Unit28, Unit29, Unit30;
{$R *.dfm}

procedure TForm1.N2Click(Sender: TObject);
begin
  Form1.1.ShowModal;
end;

procedure TForm1.N3Click(Sender: TObject);
begin
  form7.showmodal;
end;

procedure TForm1.N4Click(Sender: TObject);
begin
  form3.showmodal;
end;

procedure TForm1.N6Click(Sender: TObject);
begin
  form2.showmodal;
end;

procedure TForm1.N7Click(Sender: TObject);
begin
  form4.showmodal;
end;

```

```
end;
procedure TForm1.N8Click(Sender: TObject);
begin
form5.showmodal;
end;
procedure TForm1.N21Click(Sender: TObject);
begin
form6.showmodal;
end;
procedure TForm1.N9Click(Sender: TObject);
begin
form8.showmodal;
end;
procedure TForm1.N22Click(Sender: TObject);
begin
form9.showmodal;
end;
procedure TForm1.N10Click(Sender: TObject);
begin
form10.showmodal;
end;
procedure TForm1.N12Click(Sender: TObject);
begin
Button1.Visible:=true;
dbgrid1.Visible:=true;
form13.ShowModal;
end;
procedure TForm1.N31Click(Sender: TObject);
begin
```

```
dbgrid1.Visible:=true;
form24.showmodal;
Button1.Visible:=true;
end;
procedure TForm1.N41Click(Sender: TObject);
begin
dbgrid1.Visible:=true;
Form14.showmodal;
Button1.Visible:=true;
end;
procedure TForm1.N61Click(Sender: TObject);
begin
dbgrid1.Visible:=true;
form15.showmodal;
Button1.Visible:=true;
end;
procedure TForm1.N81Click(Sender: TObject);
begin
dbgrid1.Visible:=true;
form28.showmodal;
Button1.Visible:=true;
end;
procedure TForm1.N71Click(Sender: TObject);
begin
dbgrid1.Visible:=true;
form16.showmodal;
Button1.Visible:=true;
end;
procedure TForm1.N91Click(Sender: TObject);
```

```
begin
dbgrid1.Visible:=true;
form29.showmodal;
Button1.Visible:=true;
end;
procedure TForm1.N51Click(Sender: TObject);
begin
dbgrid1.Visible:=true;
form17.showmodal;
Button1.Visible:=true;
end;
procedure TForm1.N101Click(Sender: TObject);
begin
dbgrid1.Visible:=true;
form18.showmodal;
Button1.Visible:=true;
end;
procedure TForm1.N121Click(Sender: TObject);
begin
dbgrid1.Visible:=true;
form19.showmodal;
Button1.Visible:=true;
end;
procedure TForm1.N131Click(Sender: TObject);
begin
dbgrid1.Visible:=true;
form20.showmodal;
Button1.Visible:=true;
end;
```

```
procedure TForm1.N141Click(Sender: TObject);
begin
dbgrid1.Visible:=true;
form22.showmodal;
Button1.Visible:=true;
end;

procedure TForm1.N151Click(Sender: TObject);
begin
dbgrid1.Visible:=true;
form23.showmodal;
Button1.Visible:=true;
end;

procedure TForm1.N23Click(Sender: TObject);
begin
dbgrid1.Visible:=true;
form25.showmodal;
Button1.Visible:=true;
end;

procedure TForm1.N111Click(Sender: TObject);
begin
dbgrid1.Visible:=true;
form26.showmodal;
Button1.Visible:=true;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
DBGrid1.Visible:=false;
button1.Visible:=false;
Form1.Query1.SQL.Clear;
```

```

end;
procedure TForm1.N14Click(Sender: TObject);
begin
Form30.showmodal;
end;
end.
unit Unit20;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, DBCtrls;
type
    TForm20 = class(TForm)
        Label1: TLabel;
        Label2: TLabel;
        Button1: TButton;
        Edit1: TEdit;
        Edit2: TEdit;
        Label3: TLabel;
        Label4: TLabel;
        Label5: TLabel;
        ComboBox2: TComboBox;
        Label6: TLabel;
        Label7: TLabel;
        DBComboBox1: TDBComboBox;
        procedure Button1Click(Sender: TObject);
        procedure FormPaint(Sender: TObject);
    private
        { Private declarations }

```

```

public
  { Public declarations }
end;

var
  Form20: TForm20;

implementation

uses Unit1, Unit17, Unit13, Unit2;
{$R *.dfm}

procedure TForm20.Button1Click(Sender: TObject);
begin
  if (dbcombobox1.Text = "") or (edit1.Text = "") or (edit2.Text = "") then
    MessageBox(Application.Handle, 'Заполните все поля', 'Ошибка', MB_OK +
    MB_ICONHAND) else
  begin
    Form1.Query1.SQL.Clear;
    Form1.Query1.SQL.Add('SELECT DISTINCT Naimenovanie_gost as
    Наименование_категории, N_klienta as Номер_клиента, FIO as ФИО_клиента,
    Z_s_b as Занят_Свобод_занят, Data_zayavki as Дата_заселения FROM гост_,
    общая, клиенты, клиенты2 ');
    Form1.Query1.SQL.Add('WHERE гост_комплекс.N_gost=общая.N_gost and
    клиенты2.N_klienta=клиенты.N_klienta and
    клиенты2.N_nomera=общая.N_nomera and
    Naimenovanie_gost="'+dbcomboBox1.Text+"' and Data_zayavki between
    '"+edit1.text+"' and '"+edit2.text+"' and Z_s_b='"+comboBox2.text+"'");
    Form1.Query1.Open;
  end;
  if Form1.Query1.RecordCount = 0 then showmessage ('Данной записи не
  существует') else
  form20.Close;

```

```
end;
procedure TForm20.FormPaint(Sender: TObject);
begin
  DBCombobox1.Items.Clear;
  form2.Query1.First;
  while not form2.Query1.Eof do
  begin
    DBCombobox1.Items.Add(
      form2.query1.Fields.FieldName('Naimenovanie_kat').AsString);
    form2.Query1.Next;
  end;
  form2.Query1.First;
end;
end.
```