

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки, специальности)

Бизнес-информатика
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Информационная система исследования текущего и перспективного развития
рынка облачных технологий»

Студент

Р.С. Кузнецов
(И.О. Фамилия)

(личная подпись)

Руководитель

Т.Г. Любивая
(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

Аннотация

Тема выпускной квалификационной работы: Информационная система исследования текущего и перспективного развития рынка облачных технологий.

Актуальность исследования определяется важностью роли и места облачных сервисов в современной информационной среде. Владение ситуацией на рынке облачных вычислений, понимание перспектив его развития позволяет грамотно планировать стратегию информационного развития компании и получать конкурентные преимущества.

Цель работы – разработка проекта информационной системы исследования текущего и перспективного развития рынка облачных технологий.

Объект исследования – рынок облачных вычислений.

Предмет исследования – информационная система исследования развития рынка облачных технологий.

Структура ВКР включает: введение, три главы, заключение, список используемой литературы.

В первой главе выполнен анализ деятельности Карельского филиала ПАО «Ростелеком». Построены функциональные модели бизнес-процессов организации «Как есть» и «Как должно быть».

Во второй главе разработана логическая модель базы данных, построены UML и ER-диаграммы.

В третьей главе разработана физическая модель базы данных, разработано программное обеспечение информационной системы, выполнено тестирование программного проекта.

Работа включает: страниц 67, рисунков 40, таблиц 7, источников 43.

Оглавление

Введение.....	4
Глава 1 Функциональное моделирование предметной области.....	7
1.1 Техничко-экономическая характеристика предметной области.....	7
1.2 Концептуальное моделирование предметной области	12
1.2.1 Разработка и анализ модели бизнес-процесса «Как есть».....	12
1.2.2 Обоснование необходимости автоматизированного варианта решения	16
1.3 Постановка задачи на разработку АИС	19
1.4 Разработка модели бизнес-процесса «Как должно быть»	22
Глава 2 Логическое проектирование АИС	25
2.1 Выбор технологии логического моделирования АИС	25
2.2 Логическая модель АИС и её описание.....	27
2.3 Информационное обеспечение АИС.....	29
2.3.1 Характеристика нормативно-справочной и входной оперативной информации	29
2.3.2 Характеристика выходной информации.....	30
2.4 Проектирование базы данных АИС	31
2.4.1 Разработка концептуальной модели данных АИС	31
2.4.2 Разработка логической модели АИС	33
2.5 Требования к аппаратно-программному обеспечению АИС	34
Глава 3 Физическое проектирование АИС.....	36
3.1 Выбор технологии разработки программного обеспечения АИС	36
3.2 Выбор СУБД АИС	41
3.3 Разработка физической модели данных АИС	42
3.4 Разработка программного обеспечения АИС	46
3.4.1 Схема взаимосвязи модулей приложения	46
3.4.2 Описание модулей приложения	48
3.5 Тестирование АИС.....	55
Заключение	60
Список используемой литературы и используемых источников.....	61
Приложение А Программный код.....	68

Введение

Современное общество очень стремительно развивается во всех областях своей жизнедеятельности. Совершенствование технологий, появление новых устройств, выполняющих новые функции, развитие продуктов и услуг, принятие решений при участии искусственного интеллекта, машинное обучение, обработка больших данных – все это реалии сегодняшнего дня. Научно-технический прогресс вносит в ежедневную, как профессиональную, так и бытовую жизнь человека, определенный комфорт, но и существенно увеличивает её темп.

Практически все современные технологии связаны с активным использованием компьютерной техники, информационных и сетевых технологий. Многие действия люди реализуют сегодня при помощи компьютеров.

Огромные массивы информации, которые обрабатываются компьютерами, постоянно увеличиваются. Соответственно – возникает проблема организации надежных хранилищ для больших объемов данных. Требования к таким хранилищам достаточно высокие, и касаются не только большой информационной емкости, но и надежности, оперативности, доступности и других важных характеристик.

Усложняется и аппаратная часть современных компьютеров, так как к ним предъявляются те же требования – обрабатывать информацию все быстрее и все в больших объемах. При столь высоких требованиях стоимость аппаратуры возрастает. Кроме того, компаниям приходится приобретать и устанавливать большое количество разнообразного прикладного программного обеспечения, что также не дешево.

Для обустройства и поддержания современной IT-инфраструктуры многим организациям приходится нести очень высокие расходы. Для размещения компьютеров и периферийных устройств нужны большие площади. Для обслуживания техники нужны профессиональные

специалисты. Для решения профессиональных задач необходимо приобретать лицензии на использование программных продуктов. Для обеспечения оперативных и качественных коммуникаций в рамках рабочего процесса необходимо поддержание и развитие сетевой инфраструктуры предприятия.

В ответ на такое ужесточение требований специалистами в сфере производства аппаратуры и программного обеспечения, а также профессионалами в сфере управления информационными технологиями предприятий было предложено гениальное в своей простоте и надежности решение – вывести большую часть IT-инфраструктуры организаций и частных лиц во внешние информационные структуры – вычислительные облака. Такие структуры, в свою очередь, могут предоставлять информационные услуги в том наборе и в том объеме, в которых потребитель нуждается в них в настоящий момент.

Объем мирового рынка облачных сервисов к концу 2020 года превысил отметку в 312 миллиардов долларов и, по оценке специалистов, будет расти на 175% ежегодно. Менеджеры каждой компании, имеющей в своей инфраструктуре вычислительную технику, должны принять взвешенное управленческое решение об использовании облачных вычислений. В то же время компании, предоставляющие населению информационные услуги, могут выиграть, предложив потребителям новый вид сервиса.

Актуальность исследования определяется важностью роли и места облачных сервисов в современной информационной среде. Владение ситуацией на рынке облачных вычислений, понимание перспектив его развития позволяет грамотно планировать стратегию информационного развития компании и получать конкурентные преимущества.

Объект исследования – рынок облачных вычислений.

Предмет исследования – информационная система исследования развития рынка облачных технологий.

Целью выпускной квалификационной работы является разработка информационной системы исследования текущего и перспективного развития рынка облачных технологий.

Задачи выпускной квалификационной работы:

- выполнить анализ деятельности Карельского филиала ПАО «Ростелеком»;
- изучить состояние рынка облачных услуг и перспективы его развития;
- рассмотреть для Карельского филиала ПАО «Ростелеком» возможность расширения сферы предложения облачных сервисов населению;
- выполнить проектирование автоматизированной информационной системы (АИС) исследования рынка облачных технологий.

Работа состоит из введения, трех глав, заключения, списка используемой литературы.

В первой главе работы представлена технико-экономическая характеристика предметной области. Рассмотрена возможность совершенствования бизнес-процессов внедрения облачных сервисов как услуги населению от Карельского филиала ПАО «Ростелеком».

Во второй главе представлено логическое проектирование информационной системы исследования текущего и перспективного развития рынка облачных технологий.

В третьей главе описано физическое проектирование информационной системы. Представлено разработанное приложение для работы с базой данных. Выполнено тестирование приложения.

Для создания базы данных использована серверная СУБД MS SQL Server. Программное обеспечение информационной системы разработано на языке программирования C# в IDE MS Visual Studio 2019.

В заключении изложены основные результаты проделанной работы.

Глава 1 Функциональное моделирование предметной области

1.1 Техничко-экономическая характеристика предметной области

Основная деятельность ПАО «Ростелеком» состоит в предоставлении населению телекоммуникационных услуг широкого профиля:

- мобильная телефонная связь;
- доступ к цифровому телевидению;
- широкополосный доступ к сети Интернет.

В процессе развития компания постоянно внедряет в производственные процессы новейшие технологии как для клиентов, так и для внутреннего потребления. Постоянно совершенствуются процессы управления организацией. Потребителям предлагаются новые услуги и новые форматы привычных сервисов.

Основные компоненты стратегии Ростелекома в фиксированном сегменте рынка представлены на рисунке 1.



Рисунок 1 – Основные компоненты стратегии Ростелекома в фиксированном сегменте рынка

В настоящее время Ростелеком является лидером на российском рынке доступа к широкополосному Интернету и предоставления услуг цифрового телевидения.

Компания лидирует на российском рынке центров обработки данных (ЦОД) и облачных услуг IaaS (Infrastructure As A Services – Инфраструктура как услуга) по доле рынка.

Приоритетами развития организации являются:

- «умная стратегия»;
- устойчивое развитие как приоритет;
- лидерство в сфере цифровизации в масштабах страны.

Ключевые финансовые показатели деятельности организации за два последних года представлены в таблице 1 [32].

Таблица 1 – Основные финансовые показатели ПАО «Ростелеком»

Показатель	Оценка		Изменения, проценты
	2019 год	2020 год	
Выручка, млрд. руб.	475,8	546,9	15,0
OIBDA ¹ , млрд. руб.	170,7	194,1	13,7
Рентабельность OIBDA, проценты	35,9	45,5	-0,4
Чистая прибыль, млрд. руб.	21,9	25,3	15,9
Свободный денежный поток, млрд. руб.	30,3	22,7	-7,6
CAPEX ² , млрд. руб.	134,4	136,0	1,2
CAPEX/ Выручка, проценты	28,2	24,9	-3,4

С точки зрения развития наибольший интерес компании вызывает внедрение новейших информационных технологий, таких, как:

- удаленное управление «умными» устройствами;

¹ OIBDA – показатель, означающий операционный доход до вычета амортизационных отчислений

² CAPEX – капитальные расходы — затраты предприятия на приобретение внеоборотных (со сроком действия более 1 года) активов, а также на их модернизацию.

- управление беспилотными автомобилями и летательными аппаратами;
- автоматизированное управление в режиме реального времени;
- управление сервисами виртуальной и дополненной реальности;
- роботизация;
- получение информации от датчиков и управление в некритических системах.

Расширение объема предоставляемых и внедрение новых услуг облачных вычислений имеет в планах компании большое значение, так как перспективу их использования осознали сегодня российские компании, представляющие как крупный, так и средний и малый бизнес.

Оценка перспективы использования платформенных 5-G сервисов в ПАО «Ростелеком» представлена на рисунке 2.



Рисунок 2 – Оценка перспективы использования платформенных 5-G сервисов

Одной из сильных сторон ПАО «Ростелеком» являются единые корпоративные стандарты реализации бизнес-процессов для всех подразделений организации, расположенных во всех регионах Российской Федерации.

Карта бизнеса подразделений ПАО «Ростелеком» представлена на рисунке 3.

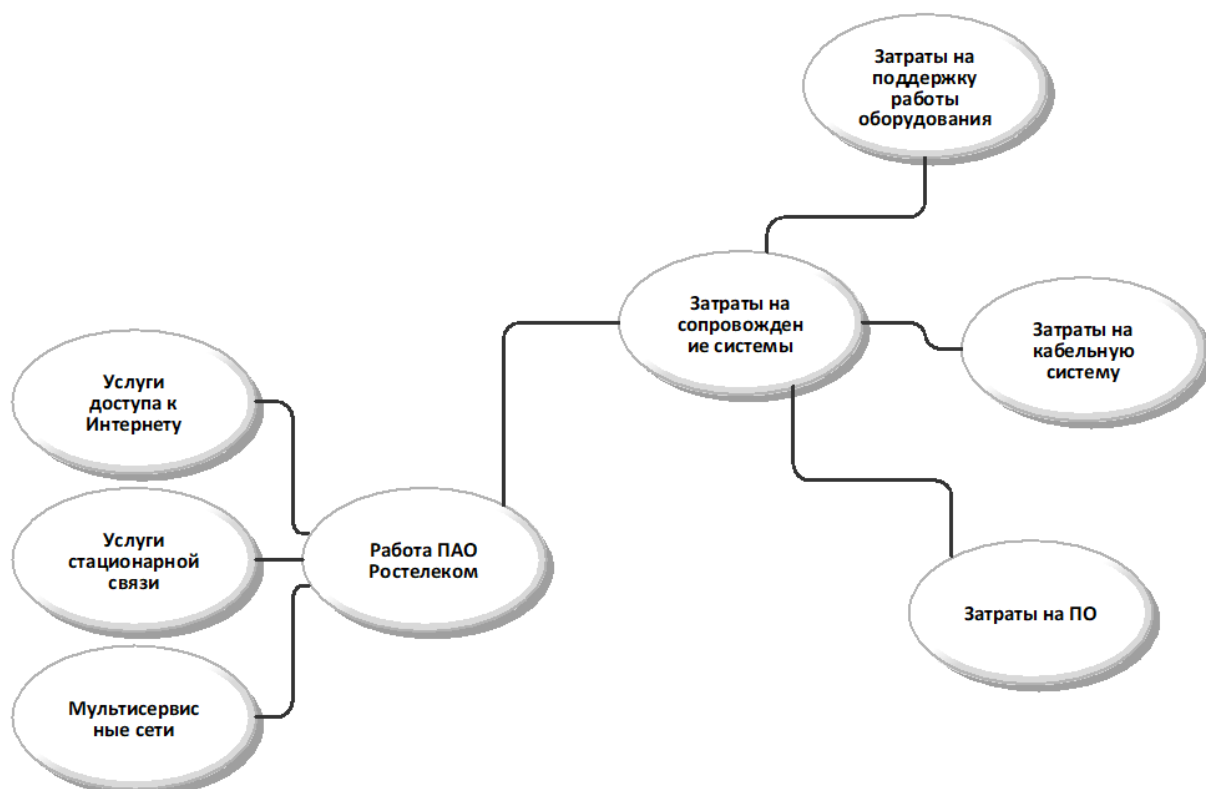


Рисунок 3 – Карта бизнеса подразделений ПАО «Ростелеком»

Все общие цели и перспективные планы развития публичного акционерного общества справедливы и для его Карельского филиала, расположенного по адресу: г. Петрозаводск, ул. Максима Горького, д. 4.

Карельский филиал ПАО «Ростелеком» – один из сети центров продаж и обслуживания, в которую входит более 800 организаций в 79 регионах Российской Федерации, находящихся в подчинении семи макрорегиональных филиалов с головным офисом в Москве.

В филиале поддерживается корпоративная культура и соблюдаются все правила ведения бизнеса, принятые в ПАО «Ростелеком».

Карельский филиал ПАО «Ростелеком» является одним из крупнейших предприятий Карелии. На базе собственной высокотехнологичной сети организация оказывает широкий спектр телекоммуникационных услуг физическим лицам, корпоративным клиентам, российским и международным операторам.

Организационная структура Карельского филиала ПАО «Ростелеком» представлена на рисунке 4.

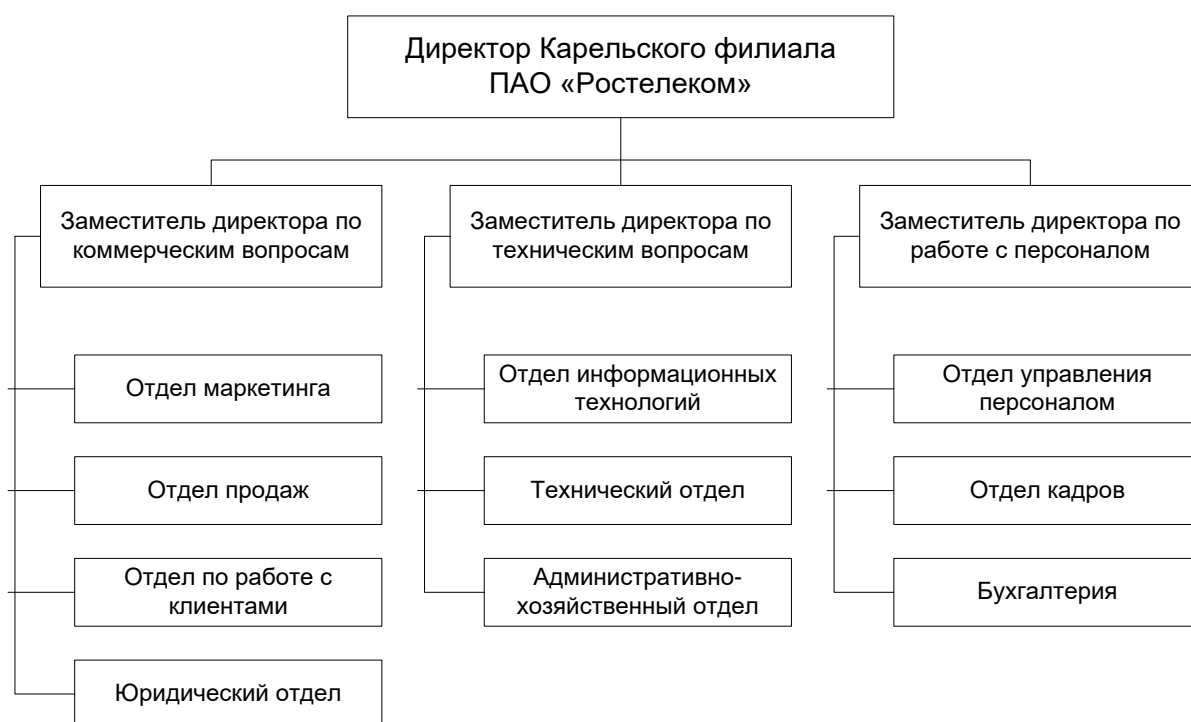


Рисунок 4 – Организационная структура Карельского филиала ПАО «Ростелеком»

В Карелии доступны населению все услуги корпорации, в том числе домашний интернет, Wink, стационарный телефон, а также мобильная связь – в общем количестве более ста услуг для дома и семьи.

Клиентам сегмента B2B доступны классическая телефония, виртуальный номер в коде 8-800, VPN, Wi-Fi, сотовая связь, облачная

телефония, информационная безопасность, видеонаблюдение и другие услуги.

1.2 Концептуальное моделирование предметной области

1.2.1 Разработка и анализ модели бизнес-процесса «Как есть»

Для разработки моделей могут быть использованы различные технологии. В работе для отображения бизнес-процессов выбраны графические нотации IDEF0 и UML.

Согласно руководящему документу РД IDEF0-2000 «Методология функционального моделирования IDEF0», методология IDEF была разработана с целью обеспечить исследование структуры, параметров и характеристик системы и состоит из трех частных методологий моделирования, основанных на графическом представлении систем [21].

Для создания функциональной модели информационной системы можно применить нотацию IDEF0, точно отображающую структуру и функции системы, потоки информации и материальных объектов, связывающих эти функции.

Выбор методологии IDEF0 для разработки модели проектируемой системы обусловлен выразительностью и полнотой графического представления процессов, простотой понимания модели представителями как заказчика информационной системы, так её разработчика, а также доступностью поддерживающих CASE-средств.

Методологию UML (Unified Modeling Language – унифицированный язык моделирования) используют как для концептуального моделирования, так и для моделирования программного обеспечения.

На рисунках 5-9 представлены диаграммы бизнес-процессов работы центров продаж и обслуживания клиентов ПАО «Ростелеком» в нотации IDEF0.

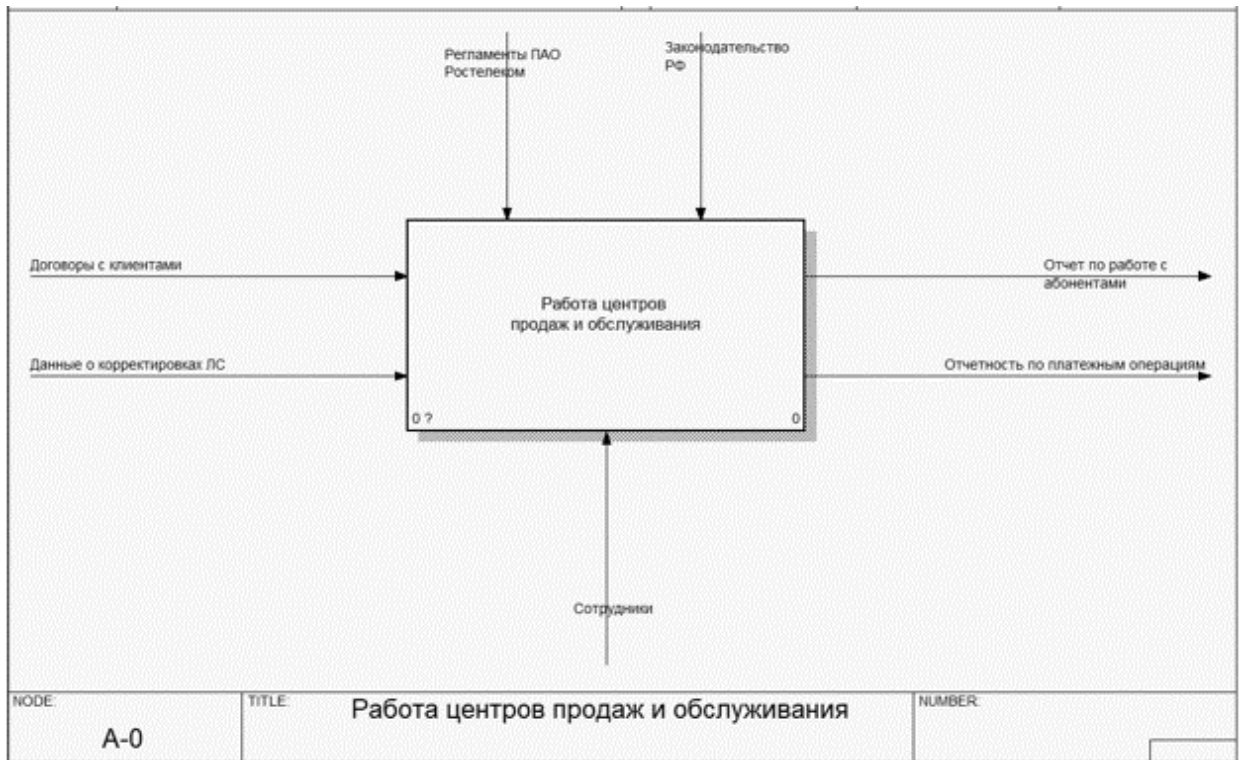


Рисунок 5 – Контекстная диаграмма

Диаграмма декомпозиции основного бизнес-процесса представлена на рисунке 6.

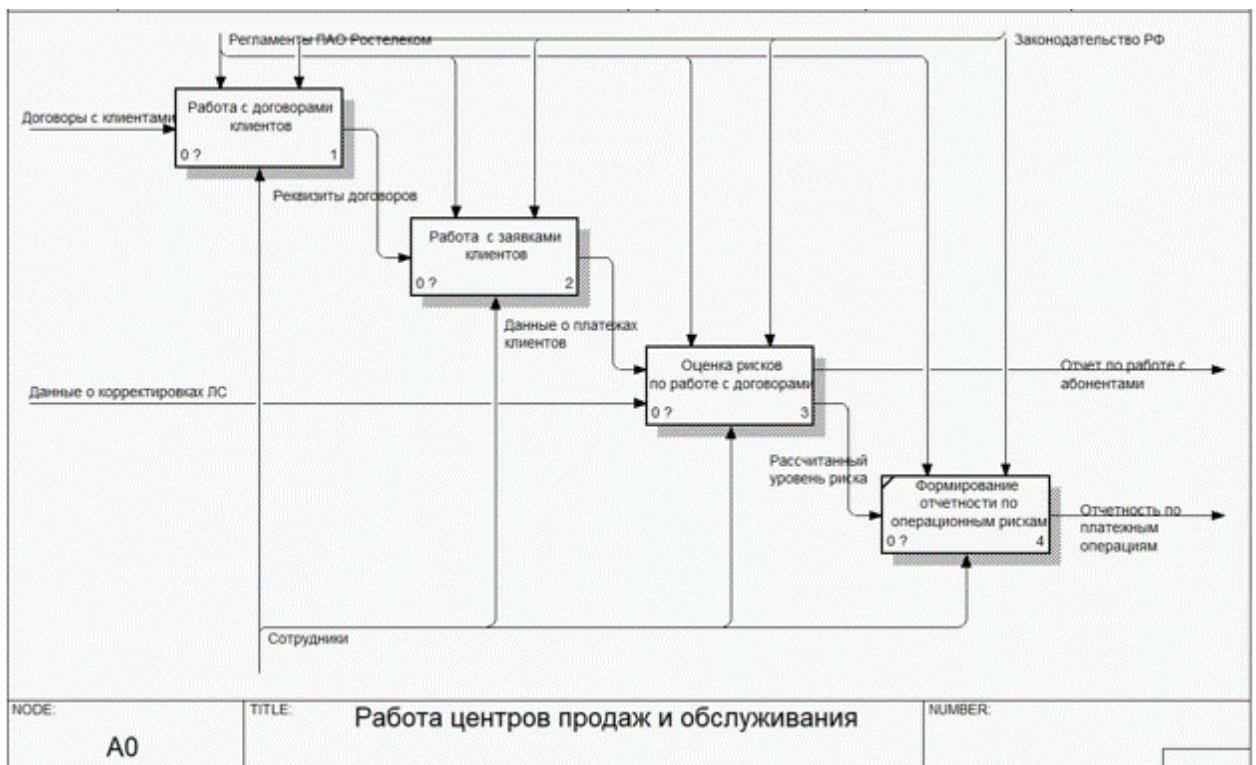


Рисунок 6 – Диаграмма декомпозиции основного бизнес-процесса

Диаграмма декомпозиции бизнес-процесса «Работа с договорами клиентов» представлена на рисунке 7.

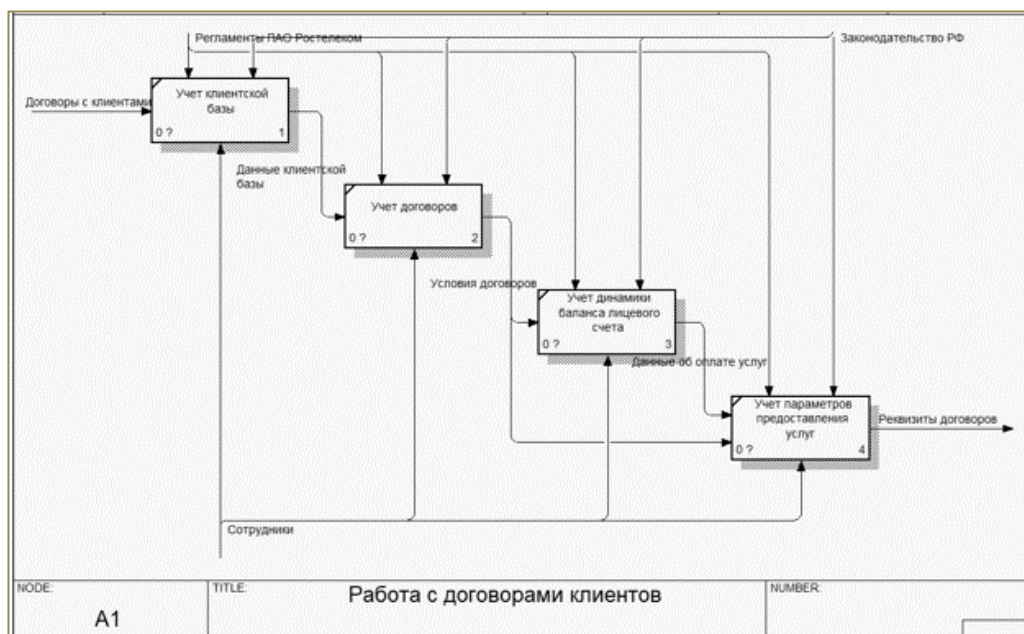


Рисунок 7 – Диаграмма декомпозиции бизнес-процесса «Работа с договорами клиентов»

Диаграмма декомпозиции бизнес-процесса «Работа с заявками клиентов» представлена на рисунке 8.

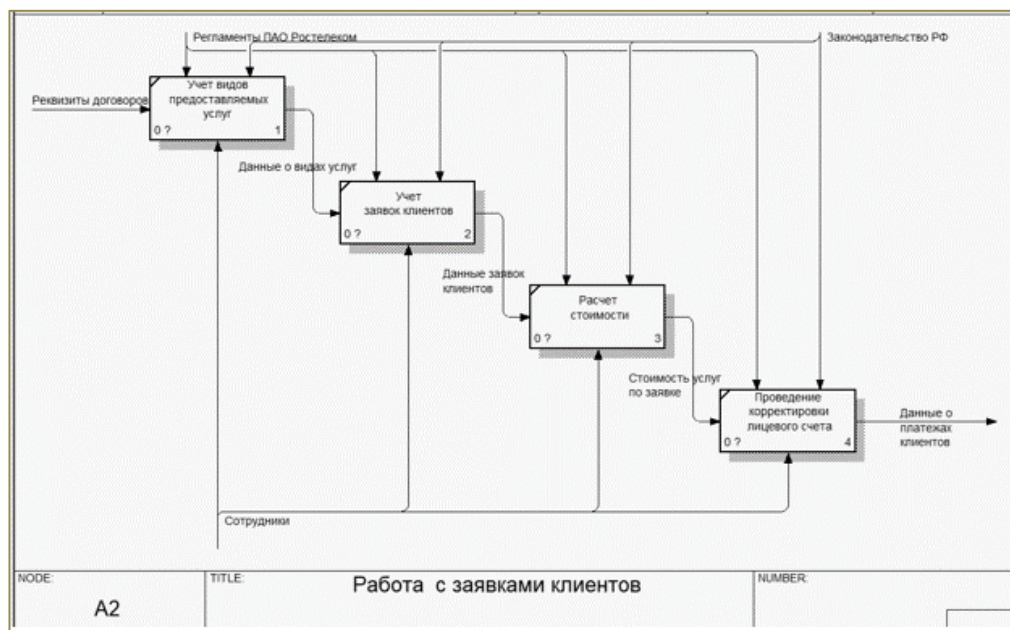


Рисунок 8 – Диаграмма декомпозиции бизнес-процесса «Работа с заявками клиентов»

Диаграмма декомпозиции бизнес-процесса «Оценка рисков по работе с договорами» представлена на рисунке 9.

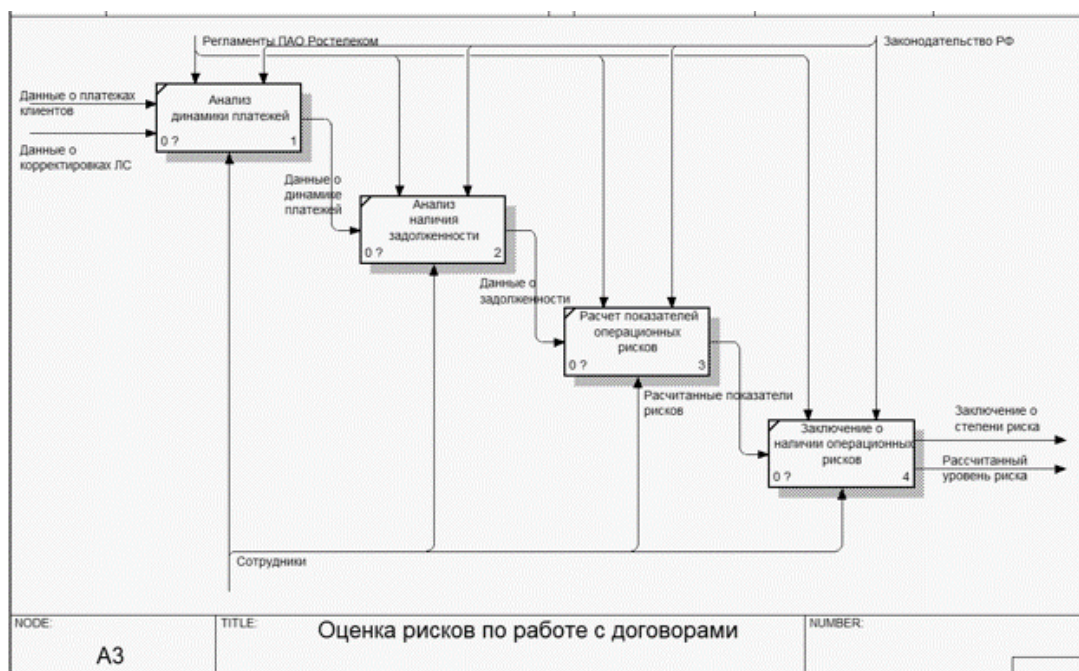


Рисунок 9 – Диаграмма декомпозиции бизнес-процесса «Оценка рисков по работе с договорами»

Диаграмма прецедентов в графической нотации UML представлена на рисунке 10.

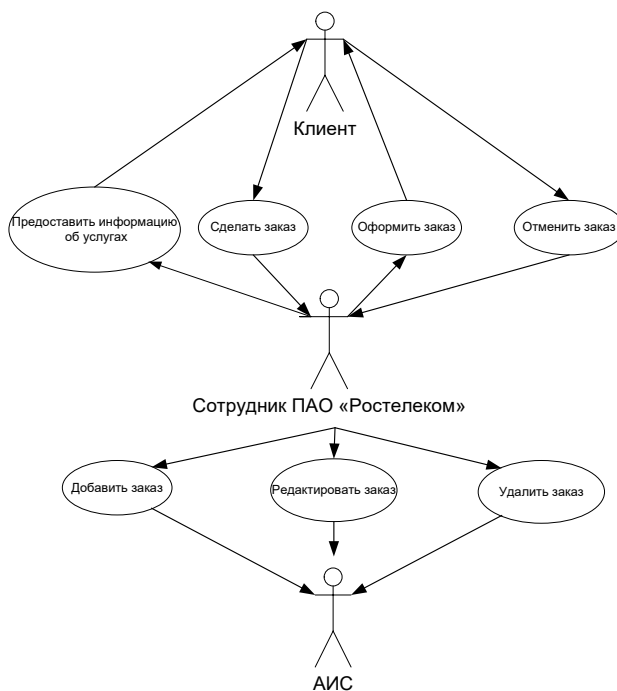


Рисунок 10 – Диаграмма прецедентов

В компании, как видно из анализа представленных моделей, используется информационная система, автоматизирующая учет клиентов, их заявок и предоставляемых им услуг.

В то же время учет развития перспективных рынков новых информационных технологий в регионах Российской Федерации не ведется, хотя для региональных представительств такое расширение имеющейся информационной системы может систематизировать деятельность по развитию филиалов в части внедрения новых услуг и стать базой для повышения их коммерческой эффективности.

В работе для проектирования справочной информационной системы, показывающей состояние рынка новых технологий и перспективы его развития, выбраны облачные технологии.

1.2.2 Обоснование необходимости автоматизированного варианта решения

Убедиться, насколько важным и необходимым является разработка автоматизированной информационной системы исследования текущего и перспективного развития облачных технологий, можно, изучив статистику использования российскими и иностранными предприятиями современных информационных технологий.

Данные для исследования получены на официальном сайте Федеральной службы государственной статистики [35]. Информация по странам была извлечена из статистического сборника «Россия и страны мира» за 2020 год (разделы 4. Валовой внутренний продукт; 9. Связь и средства информатизации).

Проведенный анализ статистических данных об использовании информационно-коммуникационных технологий, о развитии новейших компьютерных технологий позволил выдвинуть гипотезу о том, что на абсолютное значение ВВП на душу населения в 2019 году в странах мира повлияли такие показатели, как:

- число абонентов широкополосного доступа (ШПД) к сети Интернет (на 100 человек) – для исследования выбран мобильный ШПД в 2019 году;
- удельный вес населения, пользующегося сетью Интернет в 2019 году;
- доля организаций, использовавших ИКТ (облачные технологии) в 2019 году;
- доля организаций, имеющих собственный web-сайт в 2019 году.

В качестве инструмента исследования использованы электронные таблицы MS Excel, обладающие большим количеством встроенных функций и мощным аналитическим аппаратом.

Исследование проведено для 33 стран мира с использованием методов математической статистики и вычислительного аппарата Excel.

Описание переменных исследования представлено в таблице 2.

Таблица 2 – Описание переменных

Название переменной	Описание переменной
Эндогенные переменные	
ВВП	Валовой внутренний продукт в расчете на одного занятого (по паритету покупательной способности; долларов США) в 2019 году
Экзогенные переменные	
МШПД	Число абонентов широкополосного доступа к сети интернет (на 100 человек населения). Абоненты мобильного широкополосного доступа к интернету в 2019 году
ПИ	Удельный вес населения, пользующегося сетью интернет (в процентах) в 2019 году
ОС	Организации, использовавшие информационные и коммуникационные технологии в 2019 году (в процентах от общего числа организаций предпринимательского сектора). Организации, использовавшие облачные сервисы
ВС	Организации, использовавшие информационные и коммуникационные технологии в 2019 году (в процентах от общего числа организаций предпринимательского сектора). Организации, имевшие web-сайт

По результатам исследования определилась группа стран, лидирующих по всем показателям:

- Дания,
- Нидерланды,
- Финляндия,
- Швеция,
- США,
- Австралия.

Показатели по каждому параметру представлены на рисунке 11.

Цветом выделены показатели выше среднего по всем переменным.

	A	B	C	D	E	F	G
1		Страна	ВВП	МШПД	ПИ	ОС	ВС
2	1	Россия	59614	96,40	83	29	49
3	2	Австрия	115280	91,20	88	23	89
4	3	Бельгия	128842	87,40	90	40	87
5	4	Болгария	48487	91,60	68	8	51
6	5	Венгрия	71411	73,40	80	18	62
7	6	Германия	103350	87,00	88	22	88
8	7	Греция	72114	85,20	76	13	59
9	8	Дания	116608	136,90	98	56	94
10	9	Ирландия	191593	104,70	85	45	79
11	10	Испания	98016	102,10	91	22	76
12	11	Италия	104619	92,60	74	23	72
13	12	Латвия	68564	132,40	86	15	64
14	13	Литва	77363	107,80	82	23	80
15	14	Люксембург	161681	120,60	97	25	84
16	15	Нидерланды	107795	119,20	93	48	92
17	16	Норвегия	125959	103,40	98	51	78
18	17	Польша	79402	117,00	85	11	70
19	18	Португалия	75979	78,80	75	25	59
20	19	Румыния	72401	82,90	79	10	47
21	20	Словакия	75987	106,50	83	21	73
22	21	Словения	81754	83,70	83	26	83
23	22	Соединенное Королевство (Великобритания)	99245	104,00	93	42	84
24	23	Финляндия	106321	155,20	90	65	96
25	24	Франция	116402	93,60	83	19	72
26	25	Чехия	85072	92,70	81	26	83
27	26	Швеция	111954	125,70	94	57	90
28	27	Эстония	78492	157,70	90	34	81
29	28	Республика Корея	82252	113,60	17	23	65
30	29	Турция	83591	75,90	74	10	52
31	30	Япония	78829	178,90	85	59	92
32	31	Канада	99264	82,30	91	53	79
33	32	США	134190	148,90	87	52	82
34	33	Австралия	109905	129,10	87	45	77

Рисунок 11 – Показатели со значением выше среднего

В ходе проведенного анализа выявлено, что на абсолютное значение ВВП на душу населения в 2019 году в странах мира статистически значимо влияет только доля организаций, использовавших облачные технологии в 2019 году. При этом увеличение использования облачных сервисов на 1% ведет к увеличению абсолютного значения ВВП на душу населения на 9%.

Гипотеза о влиянии на абсолютное значение ВВП на душу населения в 2019 году в странах мира количества абонентов широкополосного доступа к интернету, удельного веса пользователей интернета и наличия web-сайта у организаций не подтвердилась.

На основании полученных результатов моделирования доказано, что при проведении мероприятий, направленных на увеличение абсолютного значения ВВП на душу населения, следует уделить внимание на охват использования организациями современных и перспективных облачных сервисов.

Запрос «информационная система для исследования текущего и перспективного развития рынка облачных технологий» в поисковой системе Яндекс выдал 20 миллионов результатов.

В то же время даже самые релевантные результаты дают ссылки на статьи, исследования и учебные материалы, но не на информационные системы.

Это подтверждает необходимость разработки АИС, систематизирующей имеющиеся данные для исследования текущего и перспективного развития рынка облачных технологий.

1.3 Постановка задачи на разработку АИС

Развитие информационных технологий и, в их числе облачных вычислений, происходит сегодня очень высокими темпами. Сложно уследить за их динамикой, не упустить новинки и точно понимать место устоявшихся систем и технологий, перспективы их коммерческого использования.

Организация, предоставляющая населению услуги в сфере информационных технологий, должна иметь системную картину их развития.

Разрабатываемая информационная система должна состоять из двух связанных логически, но независимых в реализации компонентов:

- база данных облачных технологий и связанных с ними объектов (центров обработки данных, предоставляющих услуги организаций);
- интерактивный отчет (дашборд), демонстрирующий состояние рынка облачных технологий и его развитие.

Требования к информационной системе разработаны в рамках модели FURPS. Эта модель понятна как заказчикам системы, так и разработчикам программного обеспечения и позволяет сформировать перечень требований, принимаемый всеми сторонами.

Требования модели FURS:

F – функциональность,

U – удобство использования,

R – надежность,

P – производительность,

S – сопровождаемость.

Функциональные требования к АИС представлены в таблице 3.

Таблица 3 – Функциональные требования к информационной системе

Требование	Статус	Полезность	Риск	Стабильность
Формирование базы данных облачных услуг	Одобрены	Критичное	Низкий	Средняя
Ведение справочника ЦОД	Одобрены	Критичное	Средний	–
Ведение справочника основных вендоров облачных услуг	Одобрены	Критичное	Средний	–
Формирование отчетов по запросам	Предложенные	Критичное	Средний	Низкая

Нефункциональные требования к информационной системе представлены в таблице 4.

Таблица 4 – Нефункциональные требования к информационной системе

Требование	Статус	Полезность	Риск	Стабильность
U – удобство использования				
Отсутствие грамматических ошибок	Одобренные	Критичное	Низкий	Низкая
Графический интерфейс	Одобренные	Критичное	Низкий	Низкая
Удобная и понятная система навигации	Одобренные	Критичное	Низкий	Низкая
Хорошо читаемый шрифт	Одобренные	Критичное	Низкий	Низкая
R – надежность				
Доступ к данным должен быть обеспечен только посредством встроенных в приложение форм	Одобренные	Критичное	Низкий	Низкая
Время восстановления системы после отказа – 12 час	Одобренные	Критичное	Средний	Низкая
P – производительность				
Среднее время запуска приложения – 10 с	Одобренные	Критичное	Средний	Средняя
Среднее время предоставления результатов выполнения запросов – 7 с	Одобренные	Критичное	Средний	Средняя
Количество пользователей – максимально 1000 человек	Одобренные	Критичное	Средний	Средняя
S – сопровождаемость				
Техническая и программная поддержка системы – по договору с заказчиком	Одобренные	Критичное	Средний	Низкая
Возможность постгарантийного обслуживания	Предложенные	Критичное	Средний	Средняя

Проектируемая автоматизируемая информационная система должна отвечать всем вышеперечисленным требованиям, согласованным с заказчиком программного продукта.

1.4 Разработка модели бизнес-процесса «Как должно быть»

Для реализации информационной системы исследования текущего и перспективного развития рынка, соответствующей поставленным выше задачам, предложена функциональная модель бизнес-процесса «Как должно быть». Контекстная диаграмма основного бизнес-процесса представлена на рисунке 12.



Рисунок 12 – Контекстная диаграмма основного бизнес-процесса «Как должно быть»

База данных информационной системы может быть использована сотрудниками компании ПАО «Ростелеком» в качестве справочника о современных облачных сервисах, разнообразие которых постоянно растет, а также о вендорах, их предоставляющих и центрах обработки данных (ЦОД).

Ростелеком также развивает собственные ЦОД и сервисную платформу «Ростелеком ЦОД», но иногда территориально удобно воспользоваться

региональными центрами обработки данных, предлагающими удобные сервисные и логистические решения.

Использование проектируемой АИС наиболее целесообразно на этапе учета видов предоставляемых услуг.

Диаграмма декомпозиции бизнес-процесса «Работа с заявками клиентов» представлена на рисунке 13.

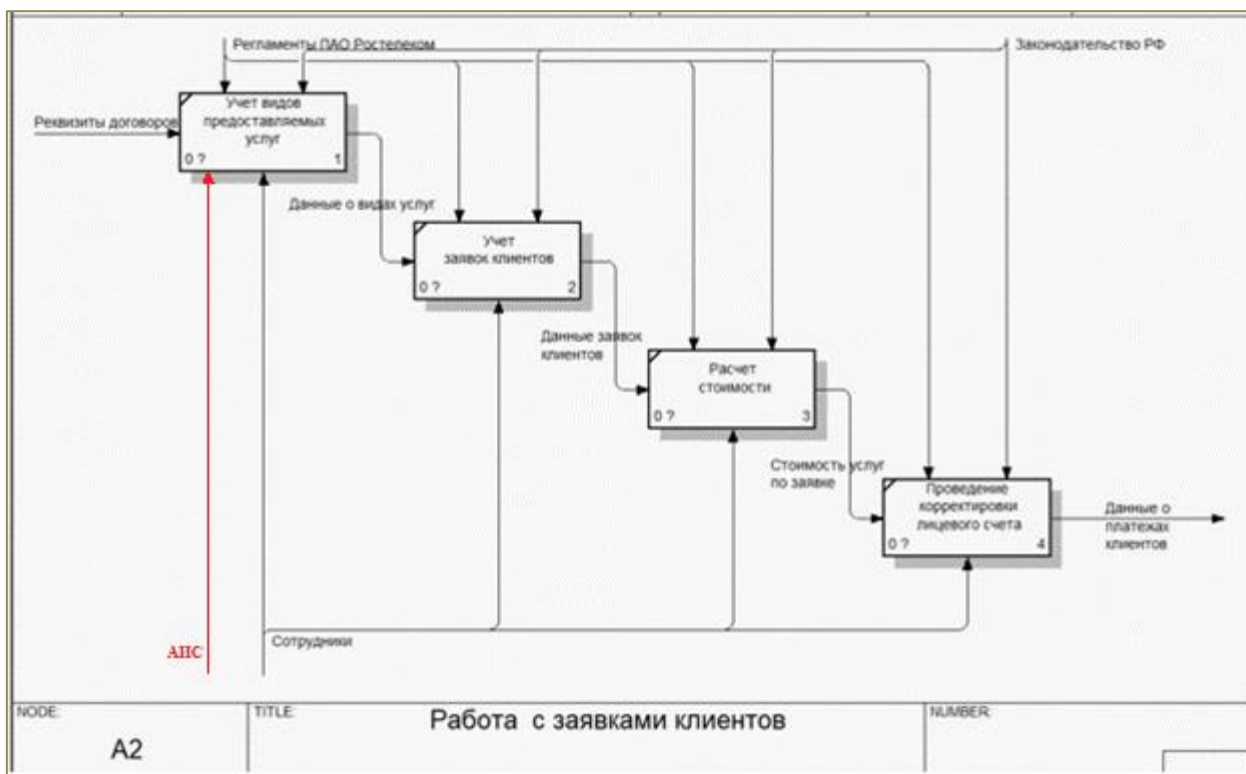


Рисунок 13 – Диаграмма декомпозиции бизнес-процесса «Работа с заявками клиентов» «Как должно быть»

Добавление в бизнес-процесс автоматизации позволит сотрудникам всех филиалов компании использовать единую информационную систему и выбирать облачные сервисы, оптимальным образом отвечающие требованиям клиентов, а также при необходимости инициировать внедрение в компании новых облачных технологий.

Выводы по первой главе

1) В процессе анализа основного бизнес-процесса в ПАО «Ростелеком» и, в силу унифицированности профессиональных процедур в компании, а также – в Карельском филиале было выяснено, что внедрение облачных вычислений является одной из приоритетных задач организации.

2) Анализ бизнес-процесса «Как есть» продемонстрировал необходимость разработки АИС, систематизирующей информацию об облачных технологиях и дающей общее представление всем сотрудникам компании единый взгляд на развитие соответствующего рынка.

3) Для наглядного представления текущего и перспективного состояния рынка облачных технологий следует разработать дашборд.

Глава 2 Логическое проектирование АИС

2.1 Выбор технологии логического моделирования АИС

Разработка автоматизированной информационной системы предполагает на начальном этапе формирование модели бизнес-процесса, который подлежит автоматизации. Моделирование предметной области необходимо для того, чтобы все участники проекта разработки АИС (заказчики и исполнители) пришли к единому пониманию назначения, функций и архитектуры системы. У каждого из участников проектной группы имеется свой взгляд на то, как должна функционировать система, и какие результаты её работы могут быть получены. Чтобы прийти к единому пониманию функциональности системы и результатов автоматизации бизнес-процесса, сначала разрабатывают и согласовывают со всеми заинтересованными лицами его модель.

Моделирование бизнес-процессов является формализованной процедурой, результатом которого считается некоторая формальная модель процесса, описанная на математическом или любом другом формализованном языке.

В отличие от описания, которое делается в свободной форме и предполагает различные трактовки, моделирование фиксирует представление бизнес-процесса с помощью общепринятых методологий, методов и нотаций, не допускающих разночтения. Таким образом, модель должна одинаково пониматься представителями как со стороны Заказчика, так и со стороны Исполнителя.

Методологией моделирования определяются системные основы исследования или проектирования, в том числе набор принципов и методов создания моделей бизнес-процессов.

Современные методы моделирования реализуют с помощью графических элементов (совокупности нотаций) и правил их использования.

Сегодня для построения моделей в различных графических нотациях разработчик может воспользоваться одним из многочисленных CASE-средств.

CASE (computer-aided software engineering)-средства представляют собой инструмент, дающий возможность реализовать процесс разработки информационной системы и/ или программного обеспечения автоматически. Основной целью применения CASE-средств является сокращение времени и затрат на разработку АИС, а также повышение их качества [21].

Нотация представляет собой систему условных знаков и правил их применения в целях описания различных категорий проектируемой информационной системы – таких объектов, как взаимосвязи процессов.

Нотации – это формализованные графические модели, которые используются, чтобы фиксировать бизнес-процессы, анализировать их и оптимизировать.

Часто CASE-средство поддерживает несколько графических нотаций, поэтому проектирование системы может быть реализовано в рамках одного выбранного продукта. Отдельные CASE-средства позволяют даже генерировать скелетный код разрабатываемой системы на одном из языков программирования (например, IBM Rational Rose).

Выбор CASE-средства во многом зависит от конкретного подхода к проектированию информационной системы. Наиболее распространенными из подходов являются:

- структурный (функциональный) подход;
- объектно-ориентированный подход;
- методология ARIS.

Для разработки модели проектируемой системы выбран объектно-ориентированный подход, так как он оптимален для описания исследуемой предметной области.

В качестве CASE-средства логического моделирования выбран универсальный графический редактор Visio компании Microsoft. Это

средство поддерживает многие нотации и позволяет разрабатывать большинство диаграмм в процессе проектирования информационной системы. В среде MS Visio можно наглядно изобразить логическую модель базы данных. Отличительной чертой выбранного CASE-средства является также простота освоения и использования.

Логическая модель системы разработана в нотации UML. Унифицированный язык моделирования (UML – Unified Modeling Language) – это семейство графических нотаций с единой метамоделью в основе.

Язык стандарта UML включает 13 типов диаграмм, отражающих различные подходы многих профессионалов к моделированию и отображающих различные аспекты системы. Диаграммы в UML определены не очень строго, допустимо присутствие элементов диаграммы одного типа в другой диаграмме. Стандарт UML предписывает, что определенные элементы обычно рисуются в диаграммах соответствующего типа, но, как показывает практика, это не является догмой.

Основное назначение UML – предоставить формальное и, в то же время, достаточно удобное и универсальное средство, позволяющее в определенной степени снизить риск расхождений в толковании спецификаций.

Диаграмма прецедентов (диаграмма вариантов использования) предназначена для наиболее общего представления функционального назначения проектируемой системы. На диаграмме вариантов использования отображаются два основных вида сущностей: действующие лица; варианты использования, между которыми устанавливаются некоторые отношения.

2.2 Логическая модель АИС и её описание

Диаграмма вариантов использования информационной системы исследования текущего и перспективного развития рынка облачных технологий представлена на рисунке 14.

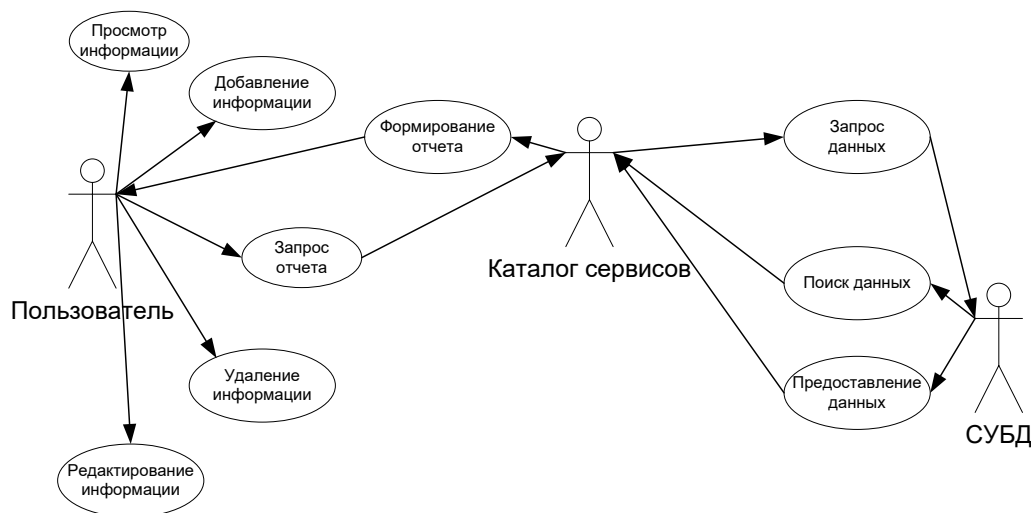


Рисунок 14 – Диаграмма вариантов использования информационной системы

Диаграмма классов является основным способом описания структуры системы. Она представляет типы объектов системы и разнообразные статические отношения между ними. Дополнительно к этому на диаграмме классов отображают свойства и операции классов, а также ограничения, относящиеся к связям между объектами. Диаграмма классов системы представлена на рисунке 15.

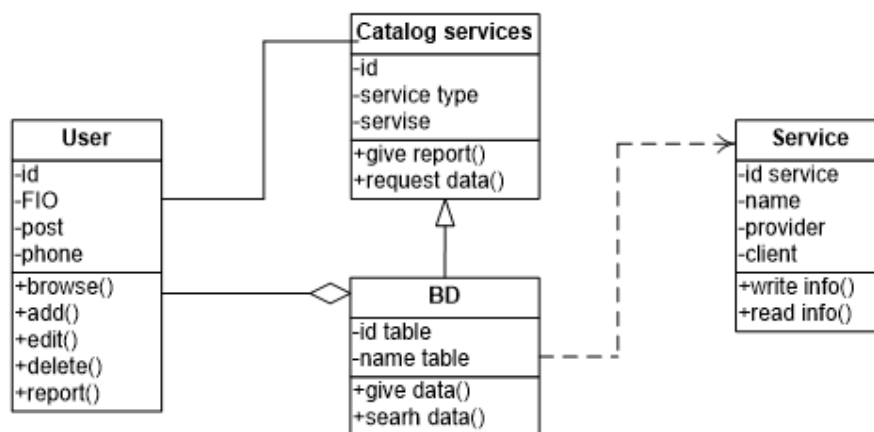


Рисунок 15 – Диаграмма классов информационной системы

Логическая модель АИС позволяет систематизировать и описать входную и выходную информацию системы.

2.3 Информационное обеспечение АИС

2.3.1 Характеристика нормативно-справочной и входной оперативной информации

В информационной системе предусмотрена входная информация, которая поступает в процессе непосредственного ввода оператора с клавиатуры.

В процессе формирования базы данных поступает следующая информация:

1) Данные о поставщиках облачных вычислений:

- наименование,
- фамилия и инициалы руководителя,
- адрес организации,
- телефон,
- центр обработки данных (дата центр).

2) Данные об облачных сервисах:

- наименование,
- описание.

3) Данные о потребителях облачных вычислений:

- наименование,
- фамилия и инициалы руководителя,
- адрес организации,
- телефон.

4) Данные о центрах обработки информации (ЦОД, Дата центр):

- наименование,
- мощности,
- страна размещения.

Входная информация поступает в систему путем добавления её в соответствующие поля таблицы. В процессе ввода корректность вводимых данных проверяется приложением.

2.3.2 Характеристика выходной информации

Проектируемая информационная система носит справочный характер и призвана наглядно показать уровень развития рынка облачных вычислений на сегодняшний момент и потенциал его развития. Поэтому в базе данных системы аккумулируются данные о потребителях и поставщиках облачных сервисов, их географическом месторасположении, размерах потребляемых услуг.

Выходная информация доступна пользователю в следующих форматах:

- экранные формы,
- отчеты по запросам,
- текстовые файлы.

С помощью экранных форм пользователю доступна вся та информация, которая вводится в систему:

1) Данные о поставщиках облачных вычислений:

- наименование,
- фамилия и инициалы руководителя,
- адрес организации,
- телефон,
- центр обработки данных (дата центр).

2) Данные об облачных сервисах:

- наименование,
- описание.

3) Данные о потребителях облачных вычислений:

- наименование,
- фамилия и инициалы руководителя,
- адрес организации,

- телефон.
- 4) Данные о центрах обработки информации (ЦОД, Дата центр):
 - наименование,
 - мощности,
 - страна размещения.

Отчеты по запросам представляют собой результат обработки запроса пользователя к базе данных. Отчет предоставляется пользователю в виде сообщения, выводимого на экран (пример представлен на рисунке 16) и дублируется в текстовый файл.

Отчет о поставщиках облачных услуг		
1. ООО «Сатурн»	Россияг. Москва, б-р Мира, д.22	+7(918)382- <u>34-52</u>
2. ПАО «Ростелеком»	Россияг. Петрозаводск, ул. Горького, д.4	+7(980)373- <u>44-44</u>

Рисунок 16 – Пример отчета по запросу

При помощи входной и выходной информации происходит обмен данными пользователя с автоматизированной информационной системой.

Важнейшей частью практически любой информационной системы является база данных.

2.4 Проектирование базы данных АИС

2.4.1 Разработка концептуальной модели данных АИС

Целью этапа логического проектирования является создание логической модели базы данных для рассматриваемой предметной области.

Для описания исследуемой предметной области выделены следующие сущности:

- Облачные сервисы – для хранения информации о предоставляемых облачных сервисах;
- Поставщики – для хранения информации о провайдерах облачных вычислений;
- Потребители – для хранения информации о потребителях облачных сервисов.

Опишем связи между сущностями.

Облачные сервисы могут оказывать несколько Поставщиков. Поставщик может предоставлять несколько Облачных сервисов. Связь «многие-ко-многим».

Облачные сервисы могут потреблять несколько Потребителей. Потребитель может потреблять несколько Облачных сервисов. Связь «многие-ко-многим».

Концептуальная ER-модель предметной области в нотации Питера Чена представлена на рисунке 17.

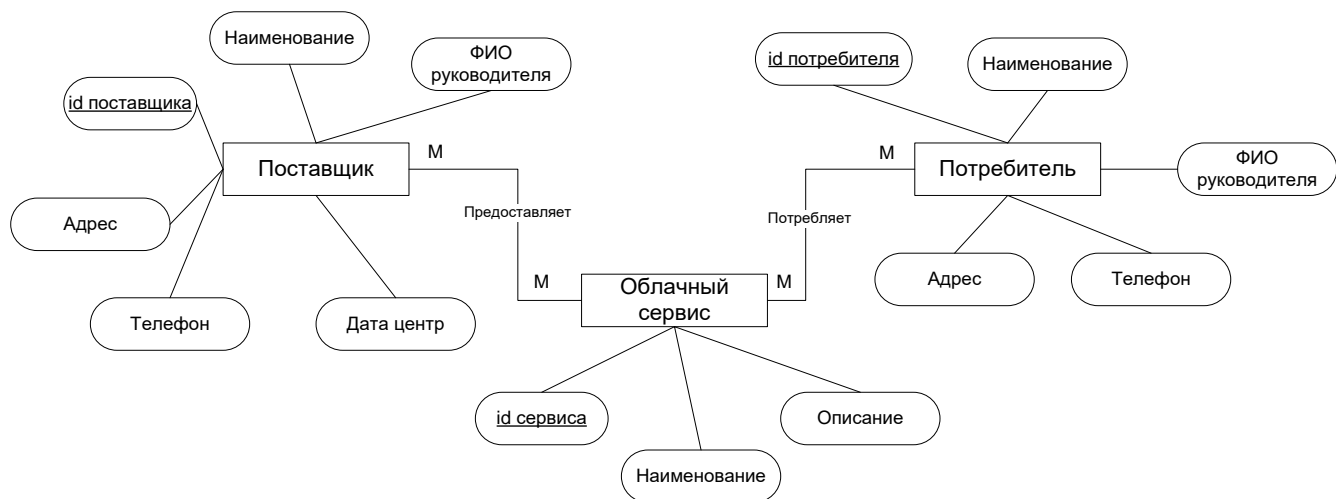


Рисунок 17 – Концептуальная ER-модель предметной области в нотации Питера Чена

Следующий этап моделирования системы – разработка логической модели базы данных.

2.4.2 Разработка логической модели АИС

Проектируемая информационная система использует реляционную модель данных.

Основными элементами реляционной модели данных являются [15]:

- отношение – реляционная таблица;
- атрибут – свойство, характеризующее сущность;
- кортеж – строка таблицы;
- домен – множество всех возможных значений определенного атрибута.

В реляционной базе данных все связи имеют вид «один-ко-многим». Для приведения связей между сущностями Облачные сервисы и Потребители, а также Облачные сервисы и Поставщики к типу «один-ко-многим» следует ввести дополнительную сущность – Услуги.

Помимо этого, потребуется ввести несколько дополнительных сущностей, уточняющих отдельные аспекты бизнес-процесса.

Логическая модель данных представлена на рисунке 18.

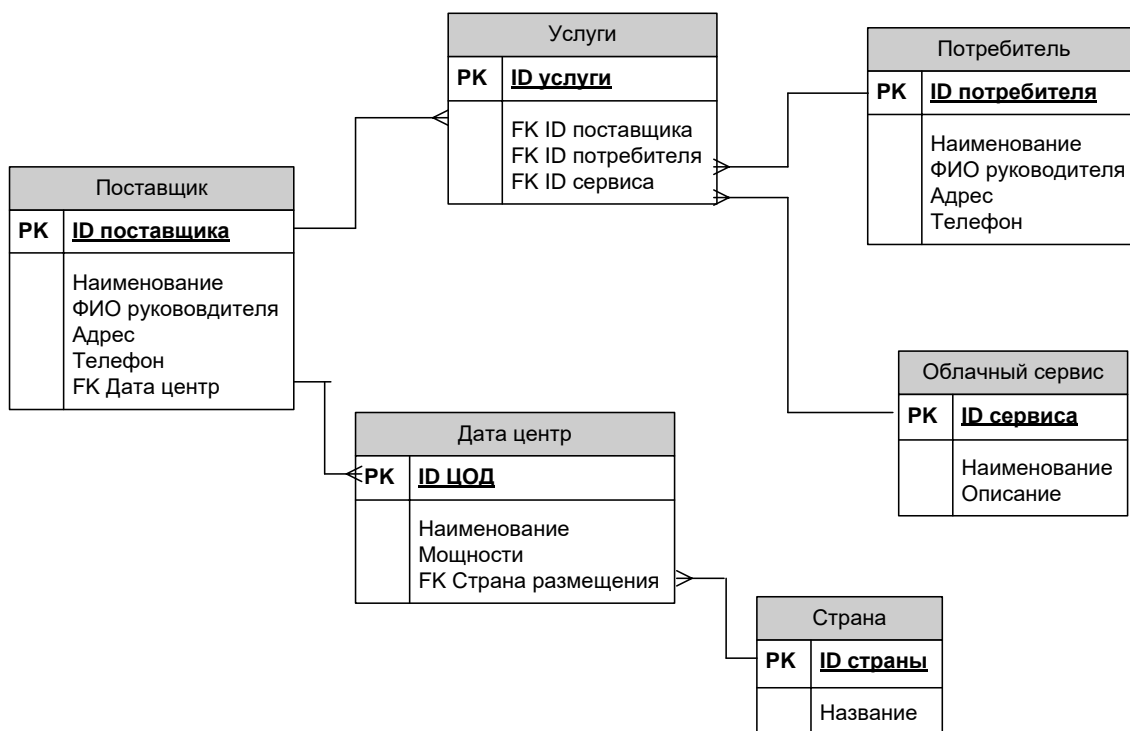


Рисунок 18 – Логическая модель данных

Разработана логическая модель нормализованной базы данных.

2.5 Требования к аппаратно-программному обеспечению АИС

Специальных требований к аппаратному и системному программному обеспечению не предусмотрено. Для эффективной работы приложения достаточно иметь персональный компьютер со следующими характеристиками:

- процессор с частотой от 2ГГц;
- объем оперативной памяти от 0,5 Гб;
- свободное дисковое пространство для хранения БД в случае установки на ПК от 256 Мб;
- материнская плата со встроенной графикой.

Среда, обеспечивающая выполнение минимальных требований для корректной работы программы:

- операционная система – Windows XP/10;
- MS Excel версии от 2013 и выше;
- совместимость с текстовыми файлами данных (часть информации выводится в файлы).

Программная документация представлена руководством пользователя и руководством программиста.

Встроенная система подсказок реализует часть программной документации путем встроенных подсказок, рекомендаций и сообщений об ошибках.

Программный код приложения открытый и полностью содержится в руководстве программиста.

Выводы по второй главе

1) Для проектирования предметной области выбран унифицированный язык моделирования UML – графическая нотация, позволяющая представить основные характеристики объекта в наглядной и понятной как для заказчика, так и для разработчика форме. Представленные диаграммы – диаграмма вариантов использования и диаграмма классов – составили основу логического представления данных.

2) Разработка концептуальной модели данных выполнена методом построения ER-диаграммы в нотации Питера Чена. Выбор обусловлен наглядностью полученной модели, которая далее положена в основу формирования логической модели данных.

3) Логическая модель базы данных информационной системы разработана на основе представленной ранее ER-диаграммы и ориентирована на реляционную модель данных. Готовая логическая модель позволяет приступить к физическому проектированию системы.

Глава 3 Физическое проектирование АИС

3.1 Выбор технологии разработки программного обеспечения АИС

Проектирование, разработка, внедрение и сопровождение информационной системы – сложный многоступенчатый процесс, каждый этап которого является стартовой площадкой для следующего. Чем качественнее и основательнее реализован этап работы над системой, тем качественнее будет готовый продукт. Если же на одной или нескольких ранних стадиях проекта допущена ошибка, то её последствия будут только расти на каждом следующем этапе работы, и на устранение погрешностей потребуется затратить много усилий и ресурсов.

Выбор архитектуры информационной системы и технологии разработки программного обеспечения является важным шагом в разработке АИС. Этот выбор во многом определяется назначением и требованиями к функциональности системы, а также планами её эксплуатации. Влияют на выбор и другие факторы – стоимость лицензии программного обеспечения, простота и доступность среды разработки и предпочтения разработчика.

Информационная система исследования текущего и перспективного развития рынка облачных технологий спроектирована как справочная система для сотрудников ПАО «Ростелеком», в частности, Карельского филиала. Это предполагает следующие допущения:

- информационная система является публичной и любой сотрудник компании может её использовать;
- приложение разрабатывается как проект, предусматривающий развитие и пополнение базы данных и аналитических инструментов инициативной группой сотрудников компании;
- система должна предоставлять пользователям актуальный срез рынка облачных вычислений на текущий момент.

Обеспечение таких позиций возможно при использовании информационной системы как локально-сетевых ресурсов. Все офисы компании «Ростелеком» оснащены локальными вычислительными сетями с выходом в Интернет, поэтому такой подход вполне оправдан. Если применение информационной системы окажется полезным для Карельского филиала организации, то можно будет масштабировать проект на другие офисы компании.

Выбор архитектурного решения проектируемой АИС состоялся как выбор между двумя популярными классами архитектуры клиент-сервер: «тонкий клиент» и «толстый клиент».

Клиент – объект, запрашивающий информацию по сети. Как правило, это персональный компьютер или рабочая станция, запрашивающая информацию у сервера.

Сервер – компьютер (как правило, высокопроизводительная рабочая станция), хранящий информацию, с которой работают сетевые клиенты.

Архитектура клиент-сервер – сетевое окружение, в котором управление данными осуществляется на серверном узле, а другим узлам предоставляется доступ к данным.

Принцип организации модели клиент – сервер представлен на рисунке 19.

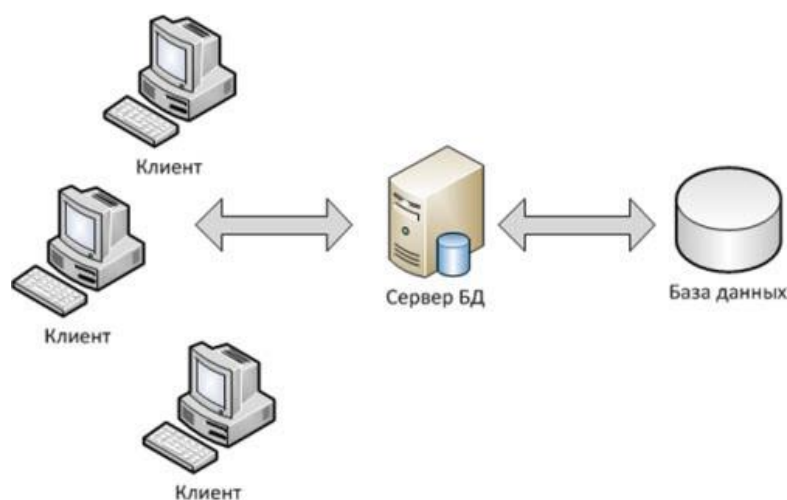


Рисунок 19 – Модель клиент-сервер

Традиционная архитектура клиент-сервер состоит из двух уровней, или звеньев: клиентского и серверного – «толстый клиент» (рисунок 20). В последние годы все большее распространение получает трехзвенная архитектура клиент-сервер, в которой прикладное программное обеспечение распределено между машинами трех типов: пользовательской машиной, промежуточным сервером и сервером базы данных – «тонкий клиент».

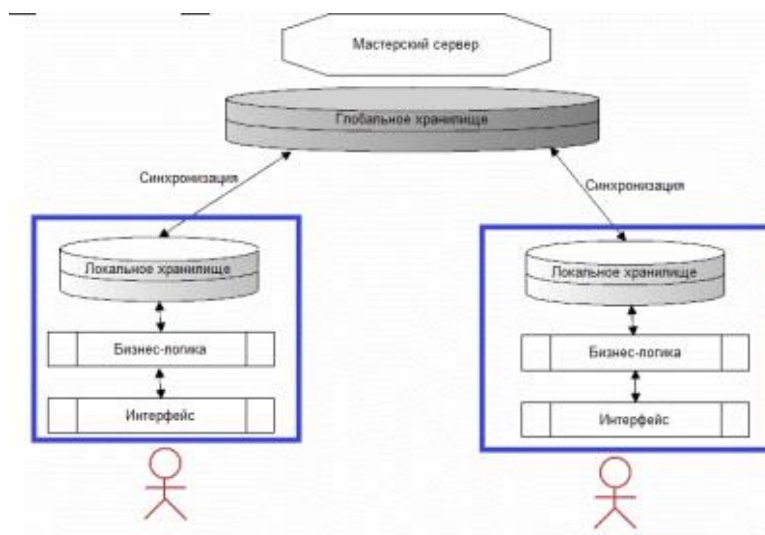


Рисунок 20 – Архитектура «толстый клиент»

Архитектура «толстый клиент» предполагает, что на сервере реализованы, главным образом, функции доступа к базам данных, а основные прикладные вычисления выполняются на стороне клиента (рисунок 21).

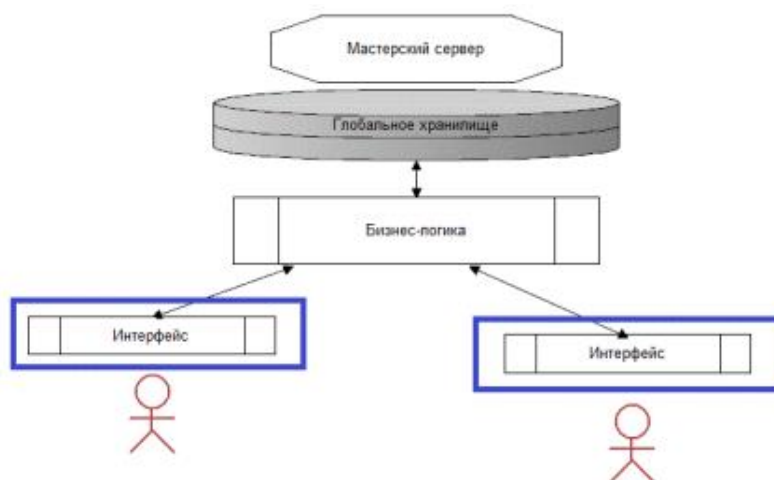


Рисунок 21 – Архитектура «тонкий клиент»

Анализ выбранных классов показал, что лучшим выбором для проектируемой АИС будет архитектура с «толстым клиентом», так как для «тонкого клиента» выделяют один недостаток, существенный для решения поставленных задач – производительность программ, написанных на языках типа PL/SQL, значительно ниже. Программы, написанные на СУБД-языках, обычно работают недостаточно надежно; ошибка в них может привести к выходу из строя всего сервера баз данных. Такие последствия можно признать катастрофическими для разрабатываемой системы, поэтому для реализации выбрана архитектура клиент-сервер класса «толстый клиент».

Современные технологии разработки программного обеспечения представляют собой систему инженерных принципов создания экономичного программного обеспечения с заданными характеристиками качества. Каждая технология разработки основывается на некоторой методологии. Несмотря на то, что программирование сегодня является мощной отраслью со множеством различных технологий и концепций разработки, сложившаяся этапность создания программного продукта практически не меняется. При разработке информационной системы реализованы все стандартные этапы разработки ПО.

Исходя из поставленных задач, для реализации системы выбран объектно-ориентированный подход. Задачи, решаемые с использованием данного подхода, описываются в терминах объектов и операций над ними, а приложение представляет собой набор объектов и связей между ними.

Системы объектно-ориентированного программирования (ООП) дают возможность визуализировать процесс создания графического интерфейса разрабатываемого приложения. Взаимодействие программных объектов между собой и их изменения описываются с помощью программного кода.

Создание программного кода в ООП базируется на использовании алгоритмических структур различных типов, исполнителями которых выступают программные объекты.

В качестве инструмента разработки приложения были рассмотрены современные универсальные языки программирования Java, Delphi и C#.

В качестве критериев сравнения были рассмотрены:

- доступность ПО;
- наличие бесплатной лицензии;
- удобство и простота среды разработки;
- популярность у разработчиков;
- поддержка ООП;
- инструменты для работы с БД;
- наличие навыков программирования у разработчика.

Каждый критерий оценивался по трехбалльной шкале:

- 0 – отсутствует (плохо);
- 1 – присутствует частично (удовлетворительно);
- 2 – полностью соответствует (хорошо).

Результаты сравнения представлены в таблице 5.

Таблица 5 – Сравнение языков программирования

Критерий	Java	Delphi	C#
доступность	1	1	2
наличие бесплатной лицензии	2	0	2
удобство и простота среды разработки	1	2	2
популярность	2	1	2
поддержка ООП	2	2	2
инструменты для работы с БД	2	1	2
наличие навыков	0	1	2
ИТОГО	10	8	14

На основе сравнительной оценки языков программирования для реализации системы был выбран язык C# и среда разработки MS Visual Studio 2019.

3.2 Выбор СУБД АИС

После выбора среды разработки и технологии программирования встает вопрос о выборе системы управления базами данных. Она должна соответствовать выбранной архитектуре клиент-сервер, быть совместима с языком программирования С# и иметь мощный инструментарий для организации запросов и их реализации. Кроме того, система управления базами данных должна поддерживать выбранную для реализации в ИС реляционную модель данных.

На современном рынке программных продуктов таким требованиям отвечают многие СУБД. Для определения, какая из современных систем управления базами данных лучше других подойдет для решения поставленных задач, были изучены и сравнены четыре наиболее популярные реляционные СУБД [24]: PostgreSQL, MS SQL Server, MySQL, Oracle.

Критерии оценки:

- доступность,
- стоимость лицензии,
- простота использования,
- ресурсоемкость,
- скорость обработки запросов,
- функциональность,
- надежность,
- полнота документации,
- наличие навыков программирования у разработчика.

Оценка критериев выполнена по трехбалльной шкале:

- 0 – отсутствует (плохо);
- 1 – присутствует частично (удовлетворительно);
- 2 – полностью соответствует (хорошо).

Результаты сравнения представлены в таблице 6.

Таблица 6 – Сравнение СУБД

Критерий	PostgreSQL	MS SQL Server	MySQL	Oracle
доступность	1	2	2	1
свободная лицензии	0	2	2	0
простота использования	0	2	1	1
ресурсоемкость	0	1	1	2
скорость обработки запросов	2	1	1	2
функционал	2	2	2	2
надежность	2	1	1	2
полнота документации	2	2	1	1
наличие навыков	1	2	1	0
ИТОГО	10	15	12	11

По результатам сравнения для реализации информационной системы следует выбрать реляционную систему управления базами данных MS SQL Server. В пользу такого выбора служит также то, что данная СУБД разработана и поддерживается компанией Microsoft и выбранная интегрированная среда разработки MS Visual Studio 2019 от того же разработчика предлагает много инструментов для работы с реляционными базами данных и полностью совместима с языком программирования C#.

3.3 Разработка физической модели данных АИС

Задача физического этапа проектирования данных заключается в выборе рациональной структуры хранения данных и методов доступа к ним, исходя из имеющегося набора методов и средств, которые представляются разработчику после этапа логического проектирования и выбора конкретной СУБД. На этапе физического проектирования осуществляется выбор формата физической записи, размещение записей в памяти и выбор метода доступа к данным.

Для разработки физической модели данных АИС воспользуемся еще одним инструментом от Microsoft – Microsoft SQL Server Management Studio. Этот инструмент позволяет наглядно сформировать структуру таблиц

реляционной базы данных и установить между ними связи. Готовую базу данных можно легко импортировать в приложение [7].

В таблице 7 представлено описание сущностей и их атрибуты.

Таблица 7 – Сущности и их атрибуты

Сущность	Атрибут	Тип данных	Описание
1	2	3	4
Provider	id	целочисленный	первичный ключ
	title	текстовый	наименование
	manager	текстовый	ФИО руководителя
	address	текстовый	адрес компании
	phone	текстовый	телефон
Users	id	целочисленный	первичный ключ
	title	текстовый	наименование
	manager	текстовый	ФИО руководителя
	address	текстовый	адрес компании
	phone	текстовый	телефон
CloudService	id	целочисленный	первичный ключ
	title	текстовый	наименование
	description	текстовый	описание
DataCenter	id	целочисленный	первичный ключ
	title	текстовый	наименование
	lv	целочисленный	уровень (1-4)
	type	текстовый	стационарный/ мобильный
	power	целочисленный	мощность
	racks	целочисленный	количество стоек
	square	целочисленный	площадь
	id_country	целочисленный	страна размещения, внешний ключ
Service	id	целочисленный	первичный ключ
	id_provider	целочисленный	внешний ключ
	id_user	целочисленный	внешний ключ
	id_service	целочисленный	внешний ключ
Country	id	целочисленный	первичный ключ
	title	текстовый	название страны

Таблицы в режиме конструктора в среде MS SQL Server Management Studio представлены на рисунках 22-27.

	Имя столбца	Тип данных	Разрешить ...
▶ 🔑	id	int	<input type="checkbox"/>
	title	nvarchar(50)	<input type="checkbox"/>
	manager	nvarchar(50)	<input checked="" type="checkbox"/>
	address	nvarchar(50)	<input checked="" type="checkbox"/>
	phone	nvarchar(50)	<input type="checkbox"/>

Рисунок 22 – Структура таблицы Provider

	Имя столбца	Тип данных	Разрешить ...
▶ 🔑	id	int	<input type="checkbox"/>
	title	nvarchar(50)	<input type="checkbox"/>
	manager	nvarchar(50)	<input checked="" type="checkbox"/>
	address	nvarchar(50)	<input checked="" type="checkbox"/>
	phone	nvarchar(50)	<input type="checkbox"/>

Рисунок 23 – Структура таблицы Users

	Имя столбца	Тип данных	Разрешить ...
▶ 🔑	id	int	<input type="checkbox"/>
	title	nvarchar(50)	<input type="checkbox"/>
	description	nvarchar(MAX)	<input checked="" type="checkbox"/>

Рисунок 24 – Структура таблицы CloudService

	Имя столбца	Тип данных	Разрешить ...
▶ 🔑	id	int	<input type="checkbox"/>
	title	nvarchar(50)	<input type="checkbox"/>

Рисунок 25 – Структура таблицы Country

	Имя столбца	Тип данных	Разрешить ...
▶	id	int	<input checked="" type="checkbox"/>
	id_provider	int	<input type="checkbox"/>
	id_user	int	<input checked="" type="checkbox"/>
	id_service	int	<input type="checkbox"/>

Рисунок 26 – Структура таблицы Service

	Имя столбца	Тип данных	Разрешить ...
▶	id	int	<input type="checkbox"/>
	title	nvarchar(50)	<input type="checkbox"/>
	lv	int	<input type="checkbox"/>
	type	nvarchar(50)	<input checked="" type="checkbox"/>
	power	int	<input checked="" type="checkbox"/>
	racks	int	<input checked="" type="checkbox"/>
	square	int	<input checked="" type="checkbox"/>
	id_country	int	<input type="checkbox"/>
	id_provider	int	<input type="checkbox"/>

Рисунок 27 – Структура таблицы DataCenter

Включенное в таблицу DataCenter поле lv (уровень) соответствует делению центров обработки данных на четыре уровня по показателю надежности:

- TIER 1 – первый уровень – не имеет резервируемых компонент, обладает одним каналом электропитания и охлаждения;
- TIER II – второй уровень – предусматривает резервированные компоненты, но обладает одним каналом электропитания и охлаждения, менее подвержен нарушениям нормального хода работы, как от плановых, так и от внеплановых действий;
- TIER III – третий уровень – имеет резервированные компоненты и несколько путей каналов для электропитания и охлаждения, но лишь один из них активен;
- TIER IV – четвертый уровень – имеет несколько активных каналов электропитания распределения нагрузки и охлаждения, предусматривает резервированные компоненты.

Между разработанными реляционными таблицами установлены связи «один-ко-многим». Схема данных базы данных DB_Cloud представлена на рисунке 28.



Рисунок 28 – Схема базы данных DB_Clouds

Физическая модель данных АИС разработана на основе логической модели. В процессе проектирования добавлены и расширены отдельные характерные для предметной области свойства объектов.

3.4 Разработка программного обеспечения АИС

3.4.1 Схема взаимосвязи модулей приложения

Реализация программного кода приложения реализована в объектно-ориентированной парадигме программирования. В объектно-ориентированных приложениях все элементы, участвующие в представлении предметной области и необходимые для решения поставленной задачи, описываются с помощью объектов, обладающих атрибутами (свойствами) и

методами (действиями). Логика поведения объектов отделяется от их внешнего представления и графического интерфейса.

Графический интерфейс приложения на языке программирования C# в IDE MS Visual Studio удобно формировать в режиме конструктора, хотя можно воздействовать на свойства объектов и программно.

Поведение объектов описывается в программном коде как описание некоторого события – например, нажатия кнопки, загрузки формы или наведения курсора мыши на какой-либо объект.

В разработанном приложении представлено семь форм. Каждая из форм несет собственную функциональную нагрузку. Логическая структура приложения с установленными между модулями связями представлена на рисунке 29.

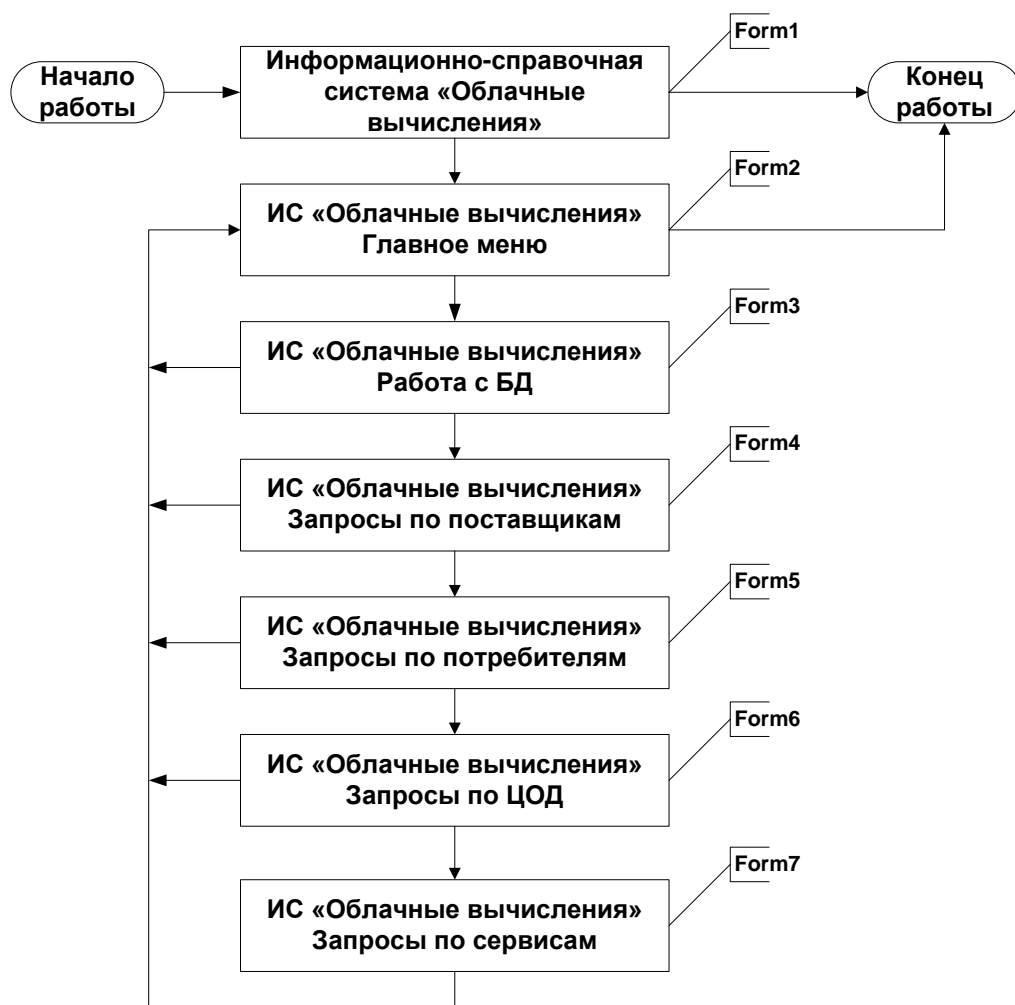


Рисунок 29 – Схема взаимосвязи модулей приложения

Связи между модулями реализованы как кнопочный интерфейс, позволяющий управлять работой приложения. Все кнопки имеют смысловые надписи, что делает навигацию по приложению простой и понятной.

3.4.2 Описание модулей приложения

Первая форма-заставка (рисунок 30) связана только со второй формой – формой главного меню (рисунок 3.13).

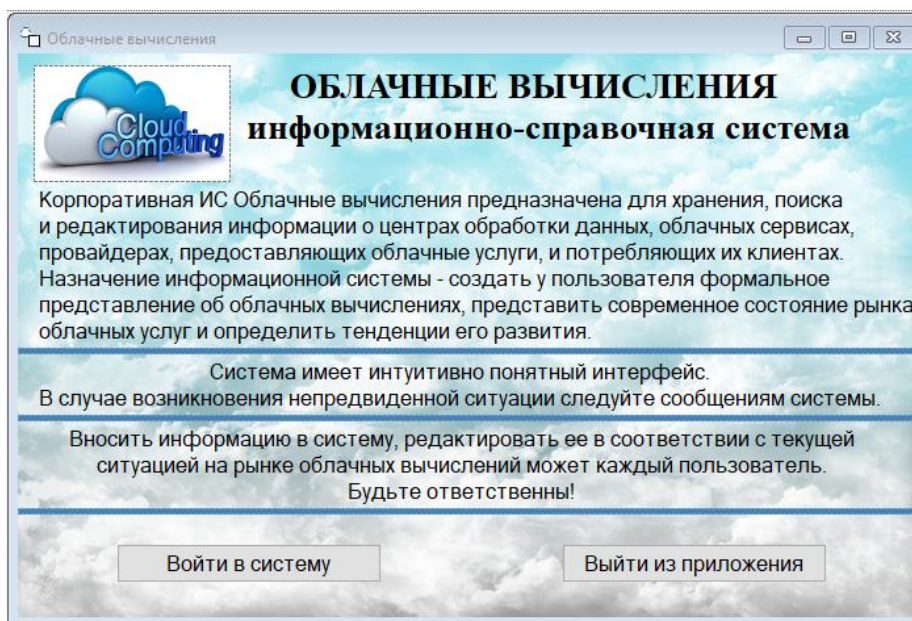


Рисунок 30 – Форма-заставка

Выйти из приложения возможно как из формы-заставки, так и из главного меню. Все остальные формы закольцованы на форму главного меню и имеют с ней как прямую, так и обратную связь.

На рисунке 31 представлена форма главного меню, на которой расположены следующие кнопки:

- Работа с БД;
- Запросы по потребителям;
- Запросы по сервисам;
- Запросы по поставщикам;
- Запросы по дата центрам;
- Выход из приложения.

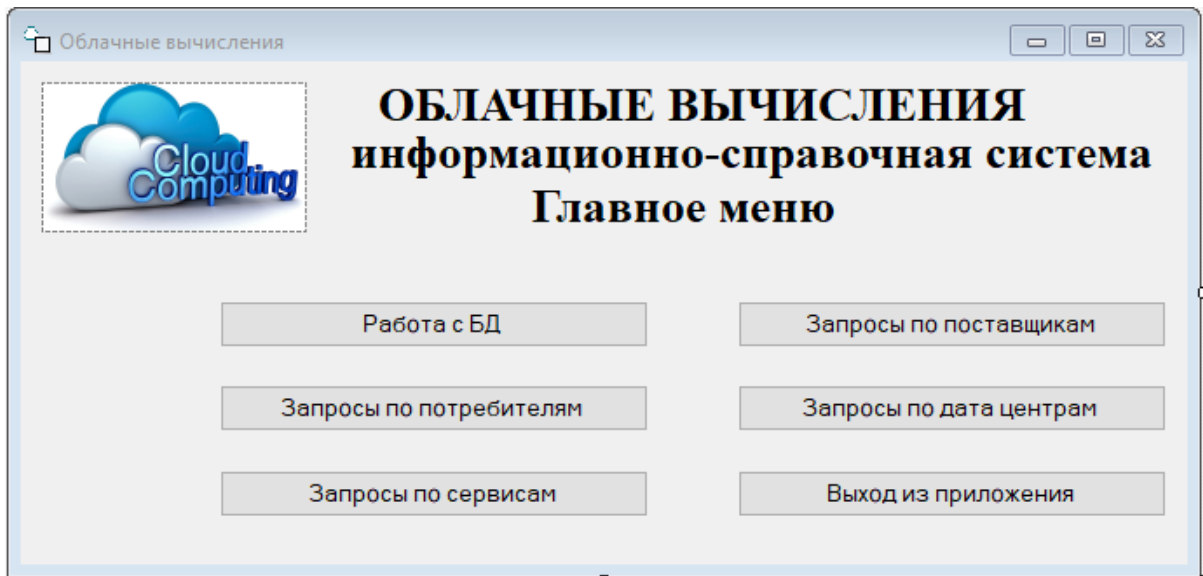


Рисунок 31 – Форма главного меню

Модуль работы с базой данных (рисунок 32) предоставляет пользователю возможность просмотра записей каждой из таблиц, добавления и удаления записей, а также редактирования имеющихся.

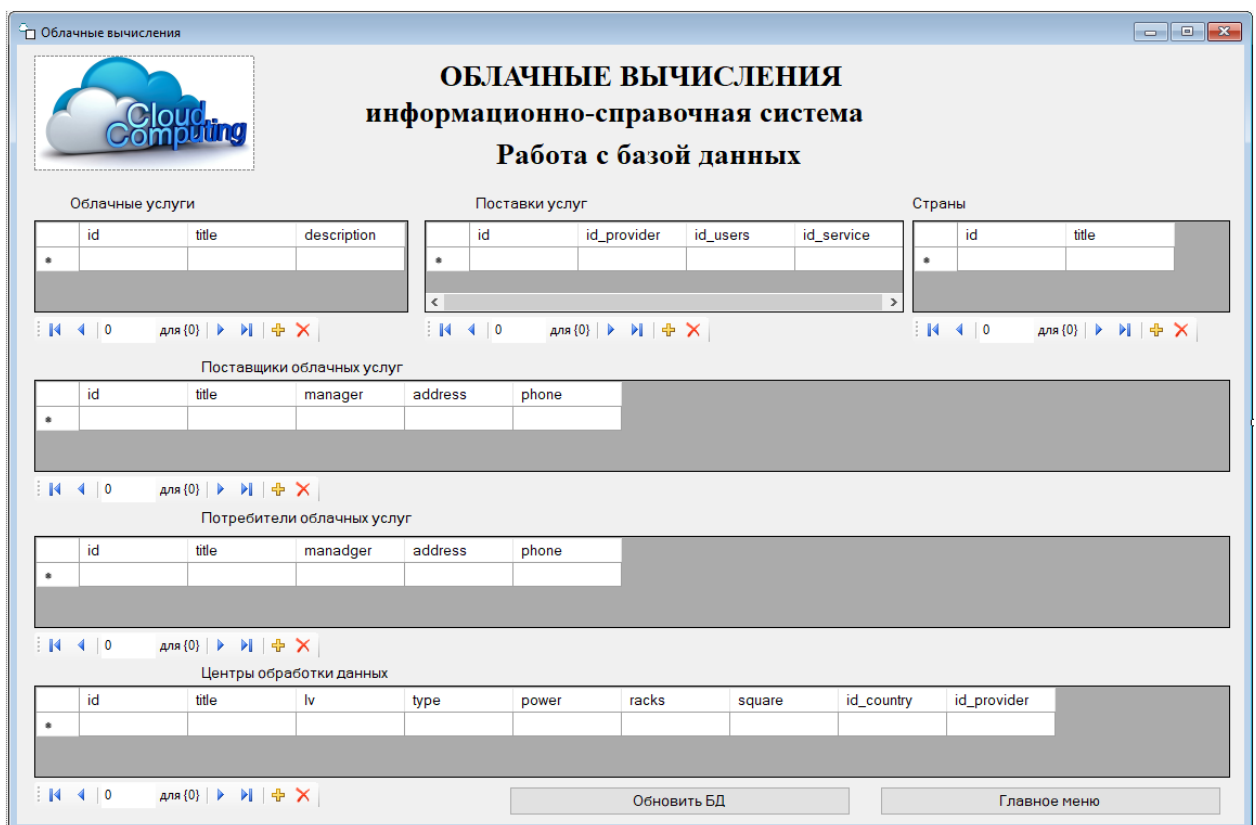


Рисунок 32 – Модуль работы с базой данных в конструкторе

Все таблицы выводятся в элемент DataGridView, связанный с соответствующей таблицей базы данных. Действиями можно управлять с помощью присоединенных к таблицам навигаторов. После внесения всех изменений следует нажать кнопку Обновить БД.

Код функции обновления базы данных приведен в листинге 1.

Листинг 1 – Код функции обновления базы данных

```
/*Обновить БД*/  
private void button2_Click(object sender, EventArgs e)  
{  
    this.dataCenterTableAdapter.Update(this.cLOADDataSet);  
    this.usersTableAdapter.Update(this.cLOADDataSet);  
    this.countryTableAdapter.Update(this.cLOADDataSet);  
    this.serviceTableAdapter.Update(this.cLOADDataSet);  
    this.cloudServiceTableAdapter.Update(this.cLOADDataSet);  
    this.providerTableAdapter.Update(this.cLOADDataSet);  
}
```

Отображение таблиц базы данных в процессе работы приложения представлено на рисунке 33.

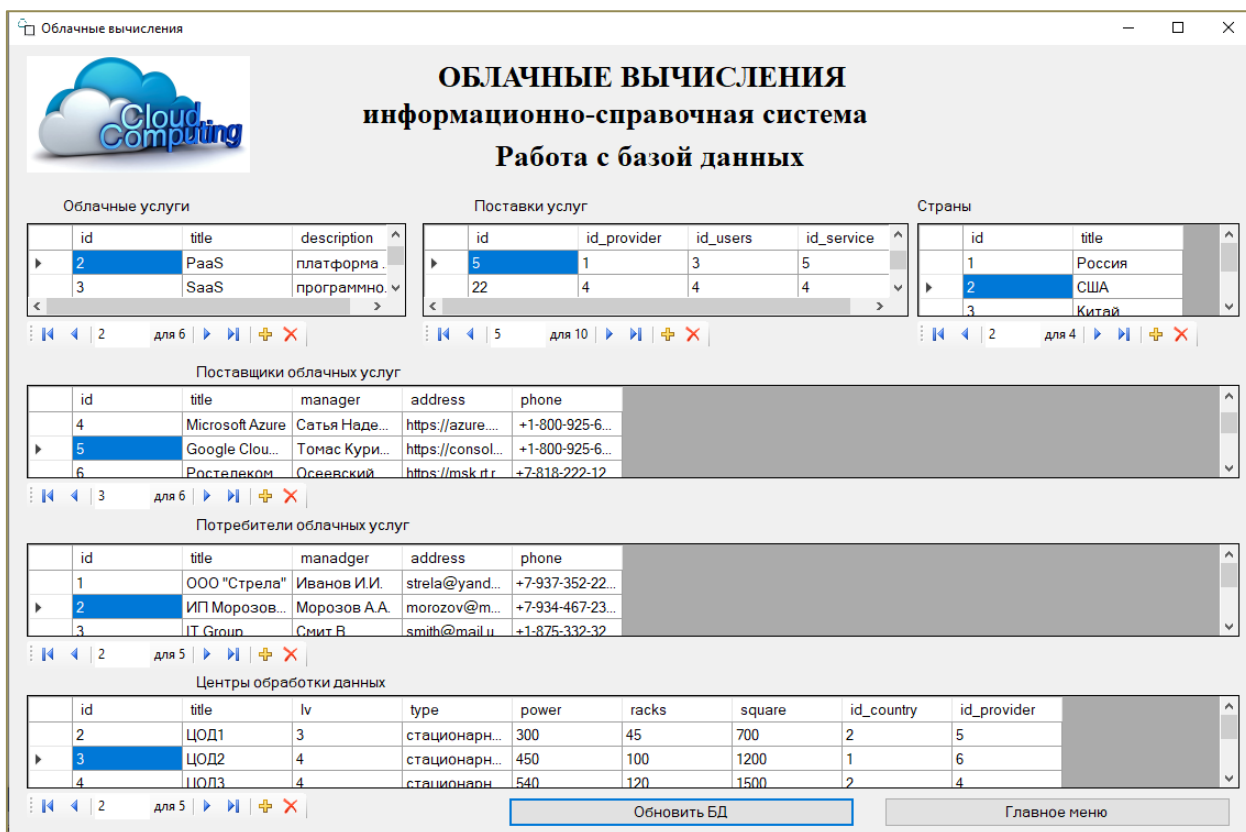


Рисунок 33 – Отображение таблиц базы данных в процессе работы приложения

На формах 4-7 (Приложение А) выводятся результаты многотабличных SQL-запросов к базе данных. Все запросы являются параметрическими.

Запрос на определение дата центров с заданным диапазоном мощностей представлен в листинге 2.

Листинг 2 – Запрос на определение дата центров с заданным диапазоном мощности

```

/*Выполнитьзапрос_Мощности*/
private void button2_Click(object sender, EventArgs e)
{
    string result; int max, min;

    //проверяем наличие данных в текстовых полях
    if (textBox3.Text=="")
    {
        MessageBox.Show("Укажите минимальную мощность","Подсказка");
    }
}

```

```

return; }
if (textBox4.Text=="")
{MessageBox.Show(" Укажите максимальную мощность", "Подсказка");
return;
}

//считываем в целочисленные переменные
min = Convert.ToInt32(textBox3.Text);
max=Convert.ToInt32(textBox4.Text);

//проверяем, чтоб минимум был меньше, иначе меняем
if (min>max)
{
int t = min; min = max; max = t;
textBox3.Text = Convert.ToString(min);
textBox4.Text = Convert.ToString(max);

//формируем запрос на SQL
result = "SELECT DataCenter.title AS 'ЦОД', DataCenter.type AS 'ТИП',
DataCenter.power AS 'Мощность', " +
" DataCenter.racks AS 'К- во стоек', Provider.title AS 'Поставщик' "+
"FROM Provider, DataCenter " +
"WHERE DataCenter.power>= " + textBox3.Text + " AND
DataCenter.power<= " + textBox4.Text+ "AND
DataCenter.id_provider=Provider.id ";

//выводим результат в dataGridView1
SqlDataAdapter dataAdapter = new SqlDataAdapter(result, sqlConnection);
DataSet dataSet = new DataSet();
dataAdapter.Fill(dataSet);

```

```

dataGridView1.DataSource = dataSet.Tables[0];

// выводим количество найденных
textBox1.Text = Convert.ToString(dataGridView1.Rows.Count-1);
}

```

Результаты выполнения запросов выводятся в DataGridView (рисунки 3.16 и 3.17) или в TextBox (рисунок 34).

Облачные вычисления

ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ
информационно-справочная система
Запросы по центрам обработки данных

Страна: Всего:

	ЦОД	ТИП	Мощность	К-во стоек	Поставщик
▶	ЦОД2	стационарный	450	100	Ростелеком
	ЦОД4	мобильный	270	34	ElasticCloud
*					

Мощность min: max: Всего:

	ЦОД	ТИП	Мощность	К-во стоек	Поставщик
▶	ЦОД1	стационарный	300	45	Google Cloud Pla...
	ЦОД2	стационарный	450	100	Ростелеком
	ЦОД4	мобильный	270	34	ElasticCloud

К-во стоек min: max: Всего:

	ЦОД	ТИП	Мощность	К-во стоек	Поставщик
▶	ЦОД1	стационарный	300	45	Google Cloud Pla...
*					

Рисунок 34 – Результаты выполнения запросов по дата центрам

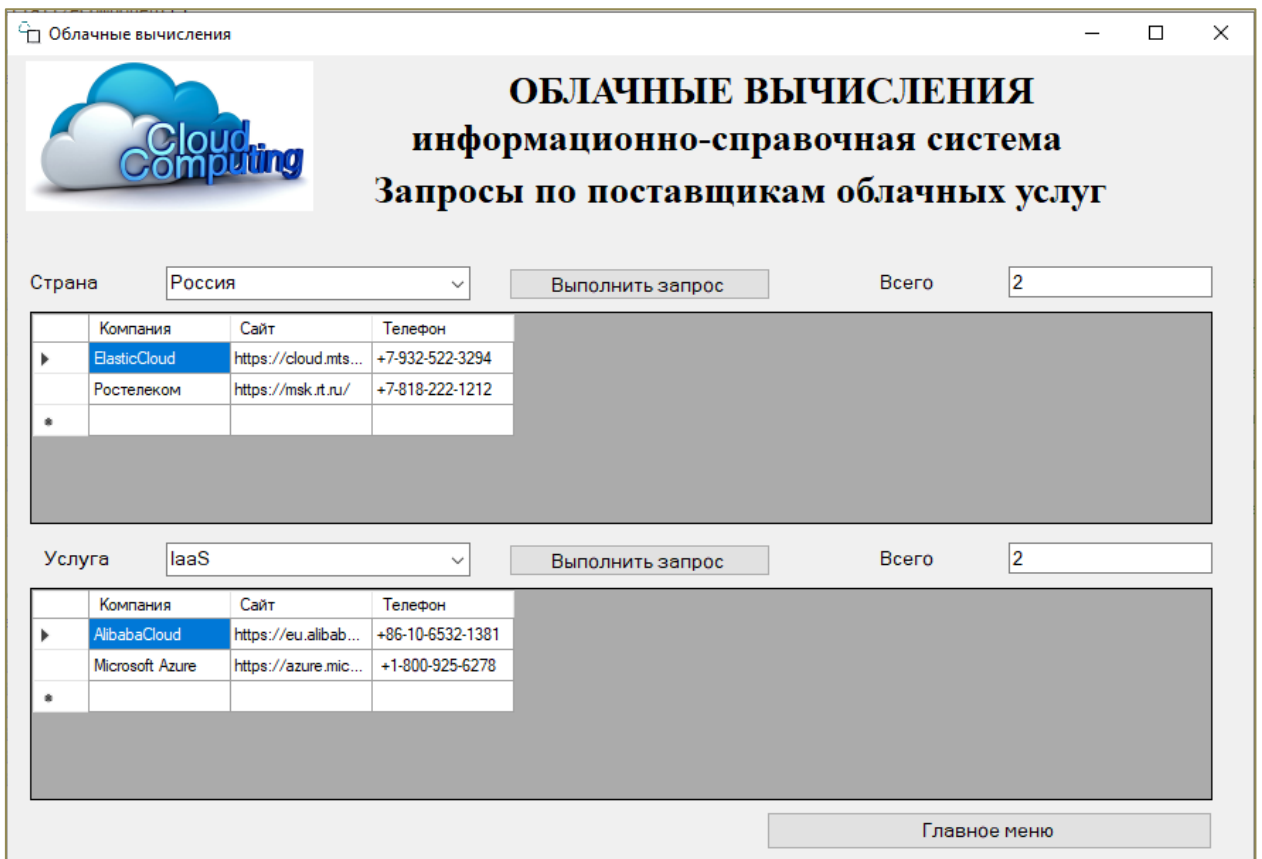


Рисунок 35 – Результаты выполнения запросов о поставщиках облачных сервисов

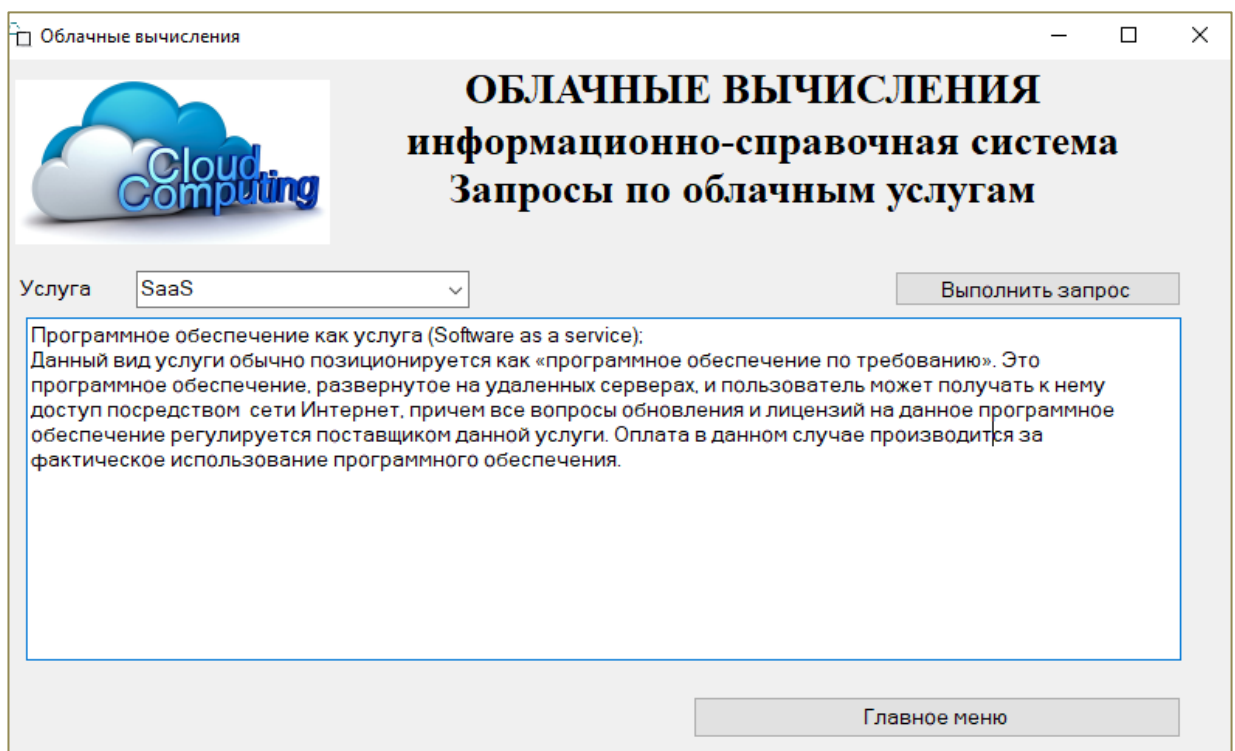


Рисунок 36 – Результат выполнения запроса выведен в текстовое поле

Аналогичным образом построены и другие модули, дающие возможность пользователю информационной системы найти и получить нужные данные.

3.5 Тестирование АИС

Тестирование разработанной информационной системы было проведено заказчиком на примере тестовой базы данных, состоящей из небольшого количества записей.

Тестирование приложения выполнялось в двух режимах:

- режим штатной работы;
- аварийный режим.

В режиме штатной работы было проверено выполнение функций системы:

- внесение информации в БД;
- удаление информации из БД;
- редактирование информации в БД;
- просмотр записей БД;
- поиск информации в БД по параметрическим запросам;
- переход по формам с помощью кнопок.

Все проверенные функции реализованы качественно и безошибочно выполняются.

Режим аварийной работы предполагает, что пользователь в процессе работы будет совершать ошибки и недопустимые с точки зрения информационной системы действия. Тестирование в аварийном режиме было проведено по трем направлениям:

- запуск функции без указания параметров её выполнения;
- ввод данных (параметров) недопустимого типа;
- ввод некорректных значений.

Все аварийные режимы работы были протестированы на примере модуля запросов по центрам обработки данных.

Первый тест – проверка реакции программы на нажатие пользователем кнопки «Выполнить запрос» (мощность) без предварительного указания минимального и максимального значений мощности ЦОД.

Результаты представлены на рисунке 37.

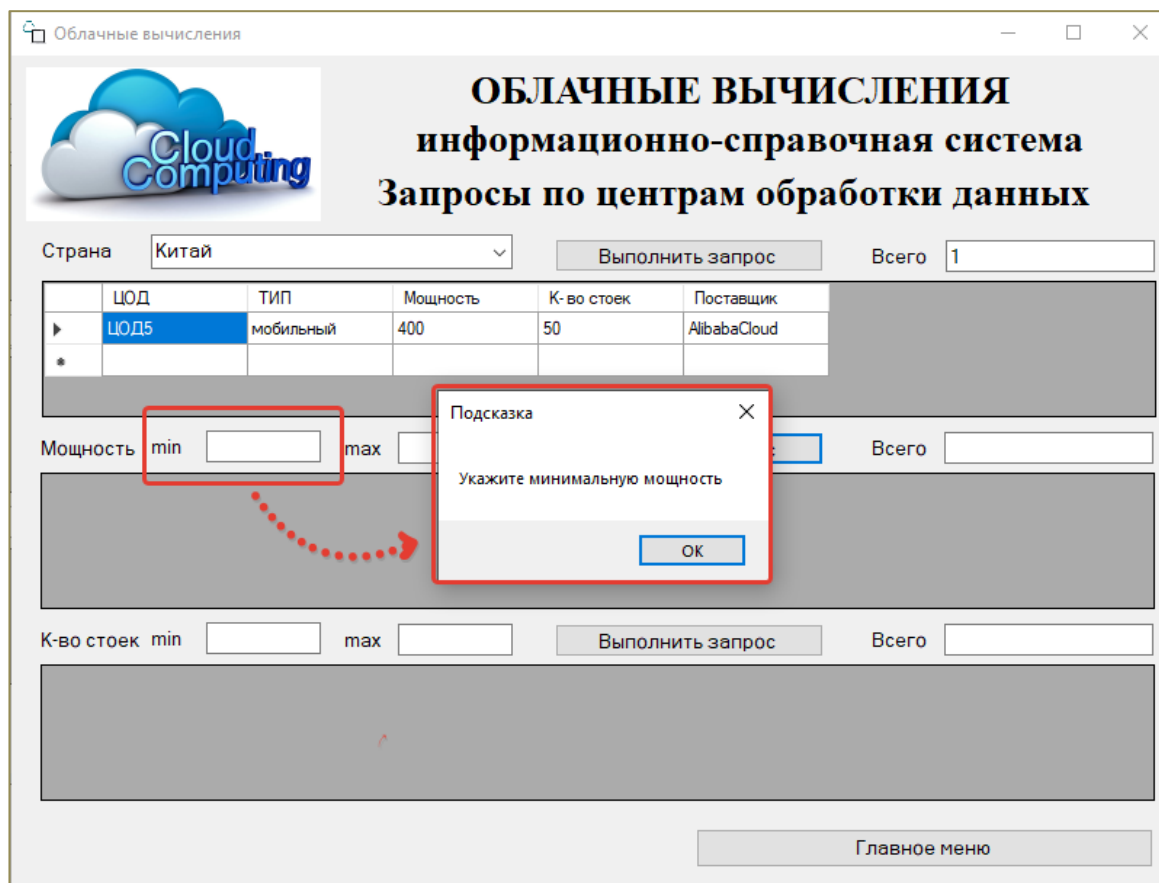


Рисунок 37 – Реакция программы на недостаток данных

Проверка на введение данных недопустимого типа была проведена на примере полей min и max (Листинг 3). По результатам тестирования выяснено, что на поля установлено ограничение ввода – допущен только ввод цифр и удаление символов

Листинг 3 – Проверка на введение данных недопустимого типа

/*Ограничения ввода в текстбоксы*/


```

private void textBox3_KeyPress(object sender, KeyPressEventArgs e)
{
    char number = e.KeyChar;
    if (!Char.IsDigit(number) && number != 8) //цифры и клавиша BackSpace
    {
        e.Handled = true;
    }
}
}

```

Под вводом некорректных значений подразумевается ситуация, когда пользователь вводит допустимые значения, но результат при этом может быть получен некорректный.

Например, пользователь вводит значение минимума больше, чем максимум. Результат тестирования представлен на рисунке 38.

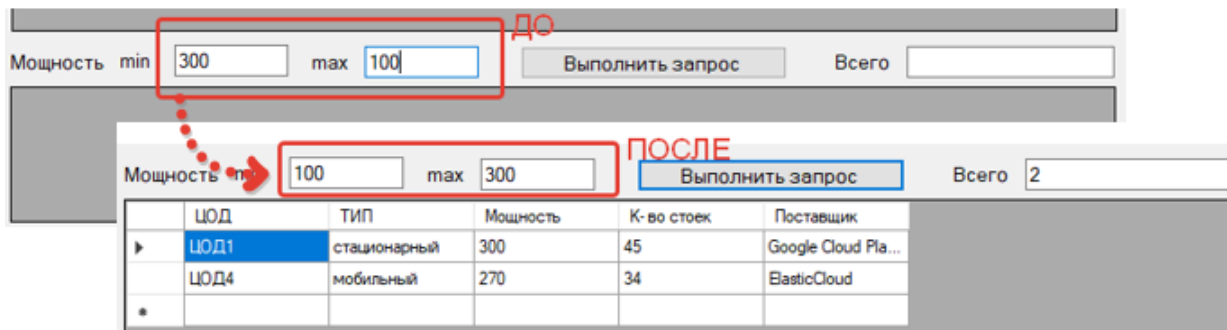


Рисунок 38 – До и после нажатия кнопки Выполнить запрос

На рисунке видно, что программа поменяла некорректно введенные пользователем значения и правильно выполнила запрос.

Тестирование показало, что все функции реализованы, приложение продемонстрировало надежную и устойчивую работу.

Информационная система предназначена для накопления информации о рынке облачных услуг и обработки её в базе данных. Со временем может быть собрана объемная полезная база данных. На сегодняшний день сложно

найти качественную выверенную статистическую информацию о развитии рынка облачных сервисов – слишком быстро все изменяется. Накопление информации в одном хранилище позволит выстроить объективную картину формирования и развития рынка.

Для визуализации данных об облачных сервисах в среде табличного процессора Excel также разработана интерактивная таблица (дашборд). Дашборд позволяет не просто визуализировать статистические данные, но также и получить динамику их изменений в интерактивном режиме, выбирая нужные параметры и сразу же получая отклик системы.

Дашборд может использоваться совместно с разработанной информационной системой для обеспечения наглядной картины текущего и перспективного развития рынка облачных технологий. Дашборд состоит из одной статической и трех динамических диаграмм.

Примеры визуализации при разных значениях параметров исследуемого рынка представлены на примере небольшого набора данных (рисунки 39 и 40).

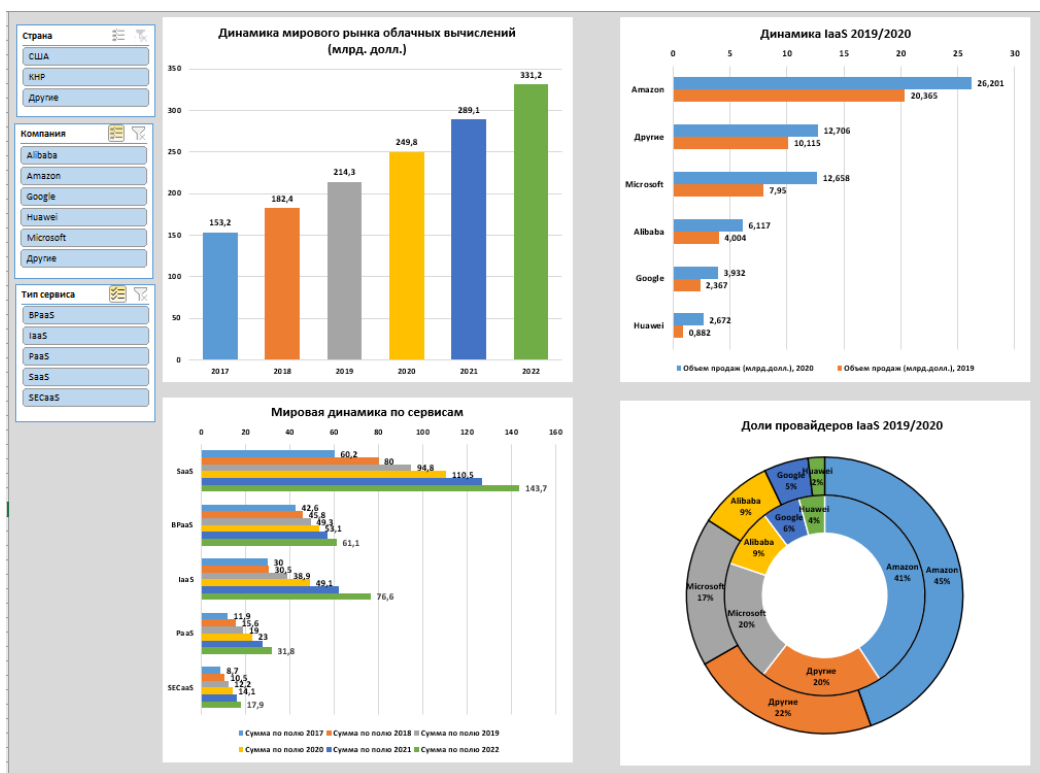


Рисунок 39 – Фильтры выключены (полная картина)



Рисунок 40 – Включены некоторые из фильтров

При работе с дашбордом при выборе нужного фильтра с помощью группы кнопок управления срезамы диаграммы перестраиваются в режиме реального времени.

Разработанная система получилась эффективной и наглядной.

Выводы по третьей главе

- 1) В качестве инструмента разработки приложения выбран язык программирования C# и среда разработки – MS Visual Studio 2019.
- 2) Физическое проектирование базы данных АИС реализовано в серверной системе управления базами данных MS SQL Server. Эта среда удобна для разработки и хорошо совместима с C#.
- 3) Разработанная информационная система была протестирована заказчиком в штатном и аварийном режимах работы. Все наборы тестов показали удобство, надежность и корректность работы системы.

Заключение

Программирование в современном мире является самостоятельной сферой человеческой деятельности, отраслью производства, в которой занято огромное количество людей. По информации Федеральной службы государственной статистики (Росстат) в России в настоящее время зарегистрировано 350 тысяч программистов, а потребность в разработчиках программного обеспечения уже сегодня составляет миллион специалистов, и эта потребность стремительно растет.

Изучение различных языков и систем программирования, освоение новых методов коммуникации с вычислительными устройствами – неотъемлемая часть жизни современного человека независимо от профессиональной принадлежности.

В ВКР на теоретическом и практическом уровнях реализованы все основные этапы разработки программного обеспечения:

- постановка технического задания;
- анализ предметной области;
- логическое и физическое проектирование базы данных;
- реализация проекта;
- отладка и тестирование готового продукта;
- передача продукта заказчику.

Разработанная в рамках ВКР информационная система исследования текущего и перспективного развития рынка облачных технологий отвечает всем заявленным требованиям заказчика. Она проста в управлении и функциональна, позволяет просматривать, редактировать, добавлять и удалять записи базы данных, а также получать результаты поисковых запросов. В дополнение к ИС в Excel разработан интерактивный инструмент визуализации статистической информации об облачных вычислениях – дашборд.

Все поставленные задачи выполнены, цель ВКР достигнута.

Список используемой литературы и используемых источников

1. Агальцов, В. П. Базы данных : в 2 книгах. Книга 2. Распределенные и удаленные базы данных [Электронный ресурс] : учебник / В.П. Агальцов. - Москва : ФОРУМ : ИНФРА-М, 2021. - 271 с. - (Высшее образование: Бакалавриат). - Режим доступа: <https://znanium.com/catalog/product/1514118>.
2. Балдин, К. В. Информационные системы в экономике [Электронный ресурс] : учебник / К. В. Балдин, В. Б. Уткин. - 8-е изд., стер. - Москва : Издательско-торговая корпорация «Дашков и К°», 2019. - 394 с. - Режим доступа: <https://znanium.com/catalog/product/1093677>.
3. Баринов, В. А. Организационное проектирование [Электронный ресурс] : учеб. : учеб. пособие для слушателей образоват. учреждений по прогр. МВА и др. прогр. подгот. упр. кадров / В. А. БариновИн-т экономики и финансов «Синергия». - Документ Bookread2. - М. : ИНФРА-М, 2017. - с. 13-65 - Режим доступа: <http://znanium.com/bookread2.php?book=492911>.
4. Бедердинова, О. И. Создание приложений баз данных в среде Visual Studio [Электронный ресурс] : учебное пособие / О.И. Бедердинова, Т.А. Минеева, Ю.А. Водовозова. - Москва : ИНФРА-М, 2021. - 94 с. - Режим доступа: <https://znanium.com/catalog/product/1243816>.
5. Беленькая, М. Н. Администрирование в информационных системах [Электронный ресурс] : учебное пособие для вузов / М. Н. Беленькая, С. Т. Малиновский, Н. В. Яковенко. - Москва : Горячая линия-Телеком, 2018. - 408 с. - Режим доступа: <https://znanium.com/catalog/product/1195564>.
6. Бизнес-планирование [Электронный ресурс] : учебник / Л. В. Бобков [и др.] под ред. Т. Г. Попадюк, В. Я. Горфинкеля. -

- Документ Bookread2. - М. : Вузов. учеб. [и др.], 2018. - 295 с. -
Режим доступа: <http://znanium.com/bookread2.php?book=360225>.
7. Бондаренко, И. С. Базы данных : создание баз данных в среде SQL Server [Электронный ресурс] : лабораторный практикум / И. С. Бондаренко. - Москва : Изд. Дом НИТУ «МИСиС», 2019. - 39 с. -
Режим доступа: <https://znanium.com/catalog/product/1232752>.
 8. Боуш, Г. Д. Методология научных исследований (в курсовых и выпускных квалификационных работах) [Электронный ресурс] : учебник / Г.Д. Боуш, В.И. Разумов. - Москва : ИНФРА-М, 2021. - 210 с. - (Высшее образование: Бакалавриат). - Режим доступа: <https://znanium.com/catalog/product/1236305>.
 9. Брежнев, Р. В. Методы и средства проектирования информационных систем и технологий [Электронный ресурс] : учебное пособие / Р. В. Брежнев. - Красноярск : Сиб. федер. ун-т, 2021. - 216 с. - Режим доступа: <https://znanium.com/catalog/product/1819341>.
 10. Варфоломеева, А. О. Информационные системы предприятия [Электронный ресурс] : учеб. пособие для вузов по направлению 09.03.03. «Приклад. информатика» и др. экон. специальностям / А. О. Варфоломеева, А. В. Коряковский, В. П. Романов. - 2-е изд., перераб. и доп. - Документ Bookread2. - М. : ИНФРА-М, 2019. - 330 с. - Режим доступа: <http://znanium.com/bookread2.php?book=1002067>.
 11. Гагарина, Л. Г. Технология разработки программного обеспечения [Электронный ресурс] : учеб. пособие для вузов по направлениям подгот. 09.04.01 и 09.03.03 «Информатика и вычисл. техника» / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Сидорова-Виснадулло под ред. Л. Г. Гагариной. - Документ Bookread2. - М. : Форум [и др.], 2018. - 400 с. : ил. - Режим доступа: <http://znanium.com/bookread2.php?book=924760>.

12. Гвоздева, В. А. Информатика, автоматизированные информационные технологии и системы [Электронный ресурс] : учеб. для студентов техн. специальностей / В. А. Гвоздева. - Документ Bookread2. - М. : ФОРУМ [и др.], 2020. - 542 с. - Режим доступа: <https://new.znaniium.com/read?id=350369>.
13. Гвоздева, Т. В. Проектирование информационных систем. Стандартизация [Электронный ресурс] : учеб. пособие / Т. В. Гвоздева, Б. А. Баллод. - Документ Reader. - СПб. [и др.] : Лань, 2019. - 250 с. - Режим доступа: <https://e.lanbook.com/reader/book/115515/#1>.
14. Гергель, В. П. Теория и практика параллельных вычислений [Текст] : учеб. пособие по приклад. математике и информатике / В. П. Гергель. - М. : Интернет-Ун-т Информ. Технологий [и др.], 2017. - 423 с. : ил., табл.
15. Голицына, О. Л. Базы данных [Электронный ресурс] : учебное пособие / О. Л. Голицына, Н. В. Максимов, И. И. Попов. - 4-е изд., перераб. и доп. - Москва : ФОРУМ : ИНФРА-М, 2020. - 400 с. - (Высшее образование: Бакалавриат). - Режим доступа: <https://znaniium.com/catalog/product/1053934>.
16. Голицына, О. Л. Информационные системы [Электронный ресурс] : учебное пособие / О. Л. Голицына, Н. В. Максимов, И. И. Попов. - 2-е изд. - Москва : ФОРУМ : ИНФРА-М, 2018. - 448 с. : ил. - (Высшее образование). - Режим доступа: <https://znaniium.com/catalog/product/953245>.
17. Григорьев, А. А. Методы и алгоритмы обработки данных [Электронный ресурс] : учебное пособие / А. А. Григорьев, Е. А. Исаев. - 2-е изд., перераб. и доп. - Москва : ИНФРА-М, 2021. - 383 с. - (Высшее образование: Бакалавриат). - Режим доступа: <https://znaniium.com/catalog/product/1032305>.

18. Дадян, Э. Г. Методы, модели, средства хранения и обработки данных [Электронный ресурс] : учебник / Э.Г. Дадян, Ю.А. Зеленков. - М. : Вузовский учебник : ИНФРА-М, 2018. - 168 с. - Режим доступа: <https://znanium.com/catalog/product/961470>.
19. Елиферов, В. Г. Бизнес-процессы: регламентация и управление [Электронный ресурс] : учебник / В.Г. Елиферов, В.В. Репин. - Москва : ИНФРА-М, 2021. - 319 с. - (Учебники для программы MBA). - ISBN 978-5-16-001825-6. - Режим доступа: <https://znanium.com/catalog/product/1241804>.
20. Заботина, Н. Н. Проектирование информационных систем [Электронный ресурс] : учебное пособие / Н. Н. Заботина. - Москва : ИНФРА-М, 2020. - 331 с. - (Высшее образование: Бакалавриат). - Режим доступа: <https://znanium.com/catalog/product/1036508>.
21. Ипатова, Э. Р. Методологии и технологии системного проектирования информационных систем [Электронный ресурс] : учебник / Э. Р. Ипатова, Ю. В. Ипатов. - 3-е изд., стер. - Москва : ФЛИНТА, 2021. - 256 с. - Режим доступа: <https://znanium.com/catalog/product/1588062>.
22. Коваленко, В. В. Проектирование информационных систем [Электронный ресурс] : учебное пособие / В.В. Коваленко. - 2-е изд., перераб. и доп. - Москва : ИНФРА-М, 2021. - 357 с. - (Высшее образование: Бакалавриат). - Режим доступа: <https://znanium.com/catalog/product/987869>.
23. Мартишин, С. А. Базы данных. Практическое применение СУБД SQL и NoSQL-типа для проектирования информационных систем [Электронный ресурс] : учебное пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. - Москва : ФОРУМ : ИНФРА-М, 2021. - 368 с. - (Высшее образование: Бакалавриат). - Режим доступа: <https://znanium.com/catalog/product/1215513>.

24. Мартишин, С. А. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench [Электронный ресурс] : учеб. пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. - М. : ИД «ФОРУМ» : ИНФРА-М, 2019. - 160 с. - (Высшее образование). - Режим доступа: <https://znanium.com/catalog/product/1007949>.
25. Медведев, М. А. Программирование на СИ# [Электронный ресурс] : Учебное пособие / Медведев М.А., Медведев А.Н., - 2-е изд., стер. - Москва :Флинта, Изд-во Урал. ун-та, 2017. - 64 с. Режим доступа: <https://znanium.com/catalog/product/948428>.
26. Мкртычев, С.В. Прикладная информатика. Бакалаврская работа : электрон. учеб.-метод. пособие / С.В. Мкртычев, О.М. Гущина, А.В. Очеповский. – Тольятти : Изд-во ТГУ, 2019.
27. Организационное проектирование: реорганизация, реинжиниринг, гармонизация [Электронный ресурс] : учеб. пособие для вузов по УГС 38.00.00 «Экономика и управление» (квалификация (степень) «магистр») / С. А. Лочан [и др.] под ред. Д. С. Петросяна. - Документ Bookread2. - М. : ИНФРА-М, 2017. - 195 с. - Режим доступа: <http://znanium.com/bookread2.php?book=773171>.
28. Основы автоматизированного проектирования [Электронный ресурс] : учеб. для вузов по направлению 09.03.01 «Информатика и вычисл. техника» (квалификация (степень) «бакалавр») / А. Н. Божко [и др.] под ред. А. П. Карпенко. - Документ Bookread2. - М. : ИНФРА-М, 2018. - 345 с. - Режим доступа: <http://znanium.com/bookread2.php?book=962578>.
29. Полищук, Ю. В. Базы данных и их безопасность [Электронный ресурс] : учебное пособие / Ю.В. Полищук, А.С. Боровский. — Москва : ИНФРА-М, 2021. - 210 с. - (Высшее образование: Специалитет). - Режим доступа: <https://znanium.com/catalog/product/1455886>.

30. Прикладная информатика: справочник [Электронный ресурс] : учебное пособие / под ред. В. Н. Волковой, В. Н. Юрьева. - Москва : Финансы и Статистика, 2021. - 767 с. - Режим доступа: <https://znanium.com/catalog/product/1831445>.
31. Программная инженерия [Текст] : учеб. для студентов вузов по направлению подгот. 231000 «Прогр. инженерия» / В. А. Антипов [и др.] под ред. Б. Г. Трусова. - М. : Академия, 2017. - 282 с. : табл., схем.
32. Ростелеком. Технологии возможностей. URL: <https://www.companu.rt.ru> (дата обращения 25.09.2021).
33. Светлов, Н. М. Информационные технологии управления проектами [Электронный ресурс] : учебное пособие / Н. М. Светлов, Г. Н. Светлова. - 2-е изд., перераб. и доп. - Москва : ИНФРА-М, 2020. - 232 с. - (Высшее образование: Бакалавриат). - Режим доступа: <https://znanium.com/catalog/product/1044525>.
34. Туккель, И. Л. Управление инновационными проектами [Электронный ресурс] : учебник / И. Л. Туккель, А. В. Сурина, Н. Б. Культин ; под ред. И. Л. Туккеля. - 2-е изд., доп. - Санкт-Петербург : БХВ-Петербург, 2020. - 416 с. - (Учебная литература для вузов). - Режим доступа: <https://znanium.com/catalog/product/1818470>.
35. Федеральная служба государственной статистики. URL: <https://rosstat.gov.ru/folder/210/document/13241> (дата обращения 17.08.2021).
36. Шишов, О. В. Технические средства автоматизации и управления [Электронный ресурс] : учебное пособие / О. В. Шишов. - Москва : ИНФРА-М, 2021. - 396 с. - (Высшее образование: Бакалавриат). - Режим доступа: <https://znanium.com/catalog/product/1157118>.
37. Шустова, Л. И. Базы данных [Электронный ресурс] : учебник / Л.И. Шустова, О.В. Тараканов. - Москва : ИНФРА-М, 2021. - 304 с. -

(Высшее образование: Бакалавриат). - Режим доступа:
<https://znanium.com/catalog/product/1362122>.

38. Яшин, В. Н. Информатика. Программные средства персонального компьютера [Электронный ресурс] : учеб. пособие для вузов по направлению «Приклад. Информатика» и др. экон. специальностям / В. Н. Яшин. - Документ Bookread2. - М. : ИНФРА-М, 2018. - 236 с. - Режим доступа: <http://znanium.com/bookread2.php?book=937489>.
39. Bruce Johnson. Professional Visual Studio 2017 : Wrox, 2017.
40. Benjamin Perkins, Jacob Vibe Hammer, Jon D. Reid. Beginning C# 7 Programming with Visual Studio 2017 : Wrox, 2018.
41. Betsy Beyer, Piotr Lewandowski, Ana Oprea, Paul Blankinship, Heather Adkins, Adam Stubblefield. Building Secure and Reliable Systems : O'Reilly Media, 2020.
42. Martin Kleppmann. Designing Data-Intensive Applications : O'Reilly Media, 2017.
43. Marek Chmel, Vladimir Muzny. Hands-On Data Science with SQL Server 2017 : Packt Publishing, 2018.

Приложение А

Программный код

Форма 1

```
namespace Cloud
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        /*****Выйти из приложения*****/
        private void button2_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
        /*****Войти в систему*****/
        private void button1_Click(object sender, EventArgs e)
        {
            Form2 frm = new Form2();
            frm.Show();
            this.Hide();
        }
    }
}
```

Форма 2

```
namespace Cloud
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }
        /*****Выход из приложения*****/
        private void button6_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
        /*****Работа с базами данных*****/
        private void button1_Click(object sender, EventArgs e)
        {
            Form3 frm = new Form3();
            frm.Show();
            this.Hide();
        }
    }
}
```

Продолжение Приложения А

```
/******Запросы по поставщикам******/
private void button2_Click(object sender, EventArgs e)
{
    Form4 frm = new Form4();
    frm.Show();
    this.Hide();
}
/******Запросы по потребителям******/
private void button3_Click(object sender, EventArgs e)
{
    Form5 frm = new Form5();
    frm.Show();
    this.Hide();
}
/******Запросы по дата центрам******/
private void button4_Click(object sender, EventArgs e)
{
    Form6 frm = new Form6();
    frm.Show();
    this.Hide();
}
/******Запросы по сервисам******/
private void button5_Click(object sender, EventArgs e)
{
    Form7 frm = new Form7();
    frm.Show();
    this.Hide();
}
}
}
```

Форма 3

```
namespace Cloud
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }
        /******Главное меню******/
        private void button1_Click(object sender, EventArgs e)
        {
            Form2 frm = new Form2();
            frm.Show();
            this.Hide();
        }
    }
}
```

Продолжение Приложения А

```
private void Form3_Load(object sender, EventArgs e)
{
    // TODO: данная строка кода позволяет загрузить данные в таблицу
    "cLOADDataSet.DataCenter". При необходимости она может быть перемещена или
    удалена.
        this.dataCenterTableAdapter.Fill(this.cLOADDataSet.DataCenter);
    // TODO: данная строка кода позволяет загрузить данные в таблицу
    "cLOADDataSet.Users". При необходимости она может быть перемещена или удалена.
        this.usersTableAdapter.Fill(this.cLOADDataSet.Users);
    // TODO: данная строка кода позволяет загрузить данные в таблицу
    "cLOADDataSet.Provider". При необходимости она может быть перемещена или удалена.
        this.providerTableAdapter.Fill(this.cLOADDataSet.Provider);
    // TODO: данная строка кода позволяет загрузить данные в таблицу
    "cLOADDataSet.Country". При необходимости она может быть перемещена или удалена.
        this.countryTableAdapter.Fill(this.cLOADDataSet.Country);
    // TODO: данная строка кода позволяет загрузить данные в таблицу
    "cLOADDataSet.Service". При необходимости она может быть перемещена или удалена.
        this.serviceTableAdapter.Fill(this.cLOADDataSet.Service);
    // TODO: данная строка кода позволяет загрузить данные в таблицу
    "cLOADDataSet.CloudService". При необходимости она может быть перемещена или
    удалена.
        this.cloudServiceTableAdapter.Fill(this.cLOADDataSet.CloudService);
}
/*****Обновить БД*****/
private void button2_Click(object sender, EventArgs e)
{
    this.dataCenterTableAdapter.Update(this.cLOADDataSet);
    this.usersTableAdapter.Update(this.cLOADDataSet);
    this.countryTableAdapter.Update(this.cLOADDataSet);
    this.serviceTableAdapter.Update(this.cLOADDataSet);
    this.cloudServiceTableAdapter.Update(this.cLOADDataSet);
    this.providerTableAdapter.Update(this.cLOADDataSet);
}
}
}
```

Форма 4

```
namespace Cloud
{
    public partial class Form4 : Form
    {
        private SqlConnection sqlConnection = null;
        public Form4()
        {
            InitializeComponent();
        }
    }
}
```

Продолжение Приложения А

```

/*****Главное меню*****/
private void button1_Click(object sender, EventArgs e)
{
    Form2 frm = new Form2();
    frm.Show();
    this.Hide();
}

private void Form4_Load(object sender, EventArgs e)
{
    // TODO: данная строка кода позволяет загрузить данные в таблицу
"cLOADDataSet.CloudService". При необходимости она может быть перемещена или
удалена.
    this.cloudServiceTableAdapter.Fill(this.cLOADDataSet.CloudService);
    // TODO: данная строка кода позволяет загрузить данные в таблицу
"cLOADDataSet.Country". При необходимости она может быть перемещена или удалена.
    this.countryTableAdapter.Fill(this.cLOADDataSet.Country);
    sqlConnection = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\CLOAD.mdf;Integrate
d Security=True;Connect Timeout=30");
    sqlConnection.Open();
}
/*****Выполнить
запрос_Страны*****/
private void button2_Click(object sender, EventArgs e)
{
    string result;
    //формируем запрос на SQL
    result = "SELECT DISTINCT provider.title AS 'Компания',provider.address AS
'Сайт', provider.phone AS 'Телефон' " +
        "FROM Provider, Country , DataCenter " +
        "WHERE Country.title=N" + comboBox1.Text + " AND
Country.id=DataCenter.id_country AND DataCenter.id_provider=Provider.id ";
    //выводим результат в dataGridView1
    SqlDataAdapter dataAdapter = new SqlDataAdapter(result, sqlConnection);
    DataSet dataSet = new DataSet();
    dataAdapter.Fill(dataSet);
    dataGridView1.DataSource = dataSet.Tables[0];
    // выводим количество найденных
    textBox1.Text = Convert.ToString(dataGridView1.Rows.Count-1);
}
/*****Выполнить запрос_Услуги*****/
private void button3_Click(object sender, EventArgs e)
{
    string result;
    //формируем запрос на SQL
    result = "SELECT DISTINCT provider.title AS 'Компания',provider.address AS
'Сайт', provider.phone AS 'Телефон' " +
        "FROM Provider, Service, CloudService " +

```

Продолжение Приложения А

```
        "WHERE CloudService.title=N" + comboBox2.Text + " AND
Service.id_service=CloudService.id AND Service.id_provider=Provider.id ";
        //выводим результат в dataGridView1
        SqlDataAdapter dataAdapter = new SqlDataAdapter(result, sqlConnection);
        DataSet dataSet = new DataSet();
        dataAdapter.Fill(dataSet);
        dataGridView2.DataSource = dataSet.Tables[0];
        // выводим количество найденных
        textBox2.Text = Convert.ToString(dataGridView2.Rows.Count-1);
    }
}
}
```

Форма 5

```
namespace Cloud
```

```
{
    public partial class Form5 : Form
    {
        private SqlConnection sqlConnection = null;
        public Form5()
        {
            InitializeComponent();
        }
        /*****Главное меню*****/
        private void button1_Click(object sender, EventArgs e)
        {
            Form2 frm = new Form2();
            frm.Show();
            this.Hide();
        }

        private void Form5_Load(object sender, EventArgs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "cLOADDataSet.CloudService". При необходимости она может быть перемещена или
            удалена.

            this.cloudServiceTableAdapter.Fill(this.cLOADDataSet.CloudService);
        }
    }
}
```


Продолжение Приложения А

```
        sqlConnection = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\CLOAD.mdf;Integrate
d Security=True;Connect Timeout=30");
        sqlConnection.Open();
    }
    /*****Выполнить запрос_Услуги*****/
    private void button3_Click(object sender, EventArgs e)
    {
        string result;
        //формируем запрос на SQL
        result = "SELECT users.title AS 'Компания', users.address AS 'Сайт',
users.phone AS 'Телефон' " +
            " FROM Users, Service, CloudService " +
            " WHERE CloudService.title=N'" + comboBox2.Text + "' AND
Service.id_service=CloudService.id AND Service.id_users=Users.id ";
        //выводим результат в dataGridView1
        SqlDataAdapter dataAdapter = new SqlDataAdapter(result, sqlConnection);
        DataSet dataSet = new DataSet();
        dataAdapter.Fill(dataSet);
        dataGridView2.DataSource = dataSet.Tables[0];
        //выводим количество найденных
        textBox2.Text = Convert.ToString(dataGridView2.Rows.Count - 1);
    }
}
}
```

Форма 6

```
namespace Cloud
{
    public partial class Form6 : Form
    {
        private SqlConnection sqlConnection = null;
        public Form6()
        {
            InitializeComponent();
        }
    }
}
```

Продолжение Приложения А

```
private void Form6_Load(object sender, EventArgs e)
{
    // TODO: данная строка кода позволяет загрузить данные в таблицу
    "cLOADDataSet.Country". При необходимости она может быть перемещена или удалена.
    this.countryTableAdapter.Fill(this.cLOADDataSet.Country);
    sqlConnection = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\CLOAD.mdf;Integrate
d Security=True;Connect Timeout=30");
    sqlConnection.Open();
}
/*****Главное меню*****/
private void button1_Click(object sender, EventArgs e)
{
    Form2 frm = new Form2();
    frm.Show();
    this.Hide();
}
/*****Выполнить
запрос_Страны*****/
private void button3_Click(object sender, EventArgs e)
{
    string result;
    //формируем запрос на SQL
    result = "SELECT DataCenter.title AS 'ЦОД', DataCenter.type AS 'ТИП',
DataCenter.power AS 'Мощность', " +
        " DataCenter.racks AS 'К- во стоек', Provider.title AS 'Поставщик' "+
        "FROM Provider, Country , DataCenter " +
        "WHERE Country.title=N" + comboBox2.Text + " AND
Country.id=DataCenter.id_country AND DataCenter.id_provider=Provider.id ";
    //выводим результат в dataGridView1
    SqlDataAdapter dataAdapter = new SqlDataAdapter(result, sqlConnection);
    DataSet dataSet = new DataSet();
    dataAdapter.Fill(dataSet);
    dataGridView2.DataSource = dataSet.Tables[0];
    // выводим количество найденных
    textBox2.Text = Convert.ToString(dataGridView2.Rows.Count-1);
}
/*****Выполнить
запрос_Мощности*****/
private void button2_Click(object sender, EventArgs e)
{
    string result; int max, min;
    //проверяем наличие данных в текстовых полях
    if (textBox3.Text=="")
    {
        MessageBox.Show(" Укажите минимальную мощность", "Подсказка");
        return;
    }
    if (textBox4.Text=="")
    {
```

Продолжение Приложения А

```
        MessageBox.Show(" Укажите максимальную мощность", "Подсказка");
        return;
    }
    //считываем в целочисленные переменные
    min = Convert.ToInt32(textBox3.Text);
    max=Convert.ToInt32(textBox4.Text);
    //проверяем, чтоб минимум был меньше, иначе меняем
    if (min>max)
    {
        int t = min;
        min = max;
        max = t;
        textBox3.Text = Convert.ToString(min);
        textBox4.Text = Convert.ToString(max);
    }

    //формируем запрос на SQL
    result = "SELECT DataCenter.title AS 'ЦОД', DataCenter.type AS 'ТИП',
DataCenter.power AS 'Мощность', " +
        " DataCenter.racks AS 'К- во стоек', Provider.title AS 'Поставщик' "+
        "FROM Provider, DataCenter " +
        "WHERE DataCenter.power>= '" + textBox3.Text + "' AND
DataCenter.power<= '" + textBox4.Text+ "'AND DataCenter.id_provider=Provider.id ";
    //выводим результат в dataGridView1
    SqlDataAdapter dataAdapter = new SqlDataAdapter(result, sqlConnection);
    DataSet dataSet = new DataSet();
    dataAdapter.Fill(dataSet);
    dataGridView1.DataSource = dataSet.Tables[0];
    // выводим количество найденных
    textBox1.Text = Convert.ToString(dataGridView1.Rows.Count-1);

    }
    /*****Выполнить запрос_К-во
стоек*****/
    private void button4_Click(object sender, EventArgs e)
    {
        string result; int max, min;
        //проверяем наличие данных в текстовых полях
        if (textBox6.Text == "")
        {
            MessageBox.Show(" Укажите минимальное количество стоек",
"Подсказка");
            return;
        }
        if (textBox7.Text == "")
        {
            MessageBox.Show(" Укажите максимальное количество стоек",
"Подсказка");
            return;
        }
    }
```

Продолжение Приложения А

```
//считываем в целочисленные переменные
min = Convert.ToInt32(textBox6.Text);
max=Convert.ToInt32(textBox7.Text);
//проверяем, чтоб минимум был меньше, иначе меняем
if (min>max)
{
    int t = min;
    min = max;
    max = t;
    textBox6.Text = Convert.ToString(min);
    textBox7.Text = Convert.ToString(max);
}

//формируем запрос на SQL
result = "SELECT DataCenter.title AS 'ЦОД', DataCenter.type AS 'ТИП',
DataCenter.power AS 'Мощность', " +
        " DataCenter.racks AS 'К- во стоек', Provider.title AS 'Поставщик' "+
        "FROM Provider, DataCenter " +
        "WHERE DataCenter.racks>= " + textBox6.Text + " AND
DataCenter.racks<= " + textBox7.Text+ "AND DataCenter.id_provider=Provider.id ";
//выводим результат в dataGridView1
SqlDataAdapter dataAdapter = new SqlDataAdapter(result, sqlConnection);
DataSet dataSet = new DataSet();
dataAdapter.Fill(dataSet);
dataGridView3.DataSource = dataSet.Tables[0];
// выводим количество найденных
textBox5.Text = Convert.ToString(dataGridView3.Rows.Count-1);
}
/*****Ограничения ввода в
текстбоксы*****/
private void textBox3_KeyPress(object sender, KeyPressEventArgs e)
{
    char number = e.KeyChar;
    if (!Char.IsDigit(number) && number != 8) // цифры и клавиша BackSpace
    {
        e.Handled = true;
    }
}

private void textBox4_KeyPress(object sender, KeyPressEventArgs e)
{
    char number = e.KeyChar;
    if (!Char.IsDigit(number) && number != 8) // цифры и клавиша BackSpace
    {
        e.Handled = true;
    }
}
```

Продолжение Приложения А

```
private void textBox6_KeyPress(object sender, KeyPressEventArgs e)
{
    char number = e.KeyChar;
    if (!Char.IsDigit(number) && number != 8) // цифры и клавиша BackSpace
    {
        e.Handled = true;
    }
}

private void textBox7_KeyPress(object sender, KeyPressEventArgs e)
{
    char number = e.KeyChar;
    if (!Char.IsDigit(number) && number != 8) // цифры и клавиша BackSpace
    {
        e.Handled = true;
    }
}
}
```

Форма 7

```
namespace Cloud
{
    public partial class Form7 : Form
    {
        private SqlConnection sqlConnection = null;
        public Form7()
        {
            InitializeComponent();
        }

        private void Form7_Load(object sender, EventArgs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "cLOADDataSet.CloudService". При необходимости она может быть перемещена или
            удалена.
            this.cloudServiceTableAdapter.Fill(this.cLOADDataSet.CloudService);
            sqlConnection = new SqlConnection(@"Data
            Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\CLOAD.mdf;Integrate
            d Security=True;Connect Timeout=30"); sqlConnection.Open(); }
    }
```

Продолжение Приложения А

```
/******Выполнить запрос******/
private void button3_Click(object sender, EventArgs e)
{
    string result;
    //формируем запрос на SQL
    result = "SELECT CloudService.description " +
        " FROM CloudService " +
        " WHERE CloudService.title=N" + comboBox2.Text + "' ";
    SqlCommand com = new SqlCommand(result, sqlConnection);
    //выводим результат в textBox;
    textBox1.Text = Convert.ToString(com.ExecuteScalar());
}
/******Главное меню******/
private void button1_Click(object sender, EventArgs e)
{
    Form2 frm = new Form2();
    frm.Show();
    this.Hide();
}
}
}
```