**Министерство образования и науки РФ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования «Ивановский государственный энергетический университет имени В.И. Ленина»**

**Кафедра программного обеспечения компьютерных систем**

**ОПЕРАТОР СОТОВОЙ СВЯЗИ**Пояснительная записка к курсовому проекту

по дисциплинам: «Проектирование и архитектура ПС», «Конструирование»

на 50 листах

**Выполнила студентка гр. 3-42 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Романова Е.А.**

**Проверили**

**к.т.н., доцент каф. ПОКС \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Игнатьев Е. Б.**

**ст. пр. ПОКС \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Фомина О.В.**

**Иваново, 2022**

# Термины, определения и сокращения

В настоящем документе использованы термины и определения, предусмотренные ГОСТ 34.003-90 [1].

Ниже приводятся сокращения, использованные в Документе:

| Сокращение | Определение |
| --- | --- |
| ИГЭУ | ФГБОУ ВПО «Ивановский государственный энергетический университет имени В.И. Ленина» |
| БД | База данных |
| ОС | Операционная система |
| ПК | Персональный компьютер |
| ПО | Программное обеспечение |
| ПОКС | Кафедра программного обеспечения компьютерных систем ИГЭУ |
| СУБД | Система управления базами данных |
| ТЗ | Техническое задание |
| ТП | Технический проект |
| MO | Оператор сотовой связи (Mobile Operator) |

# Задание

пояснительной записки к курсовому проекту «Оператор сотовой связи»

Исполнитель: Романова Е.А.

Руководители: Игнатьев Е. Б., Фомина О.В.

Документ содержит: 50 страниц, 12 рисунков, 8 таблиц

Ключевые слова: СОТОВАЯ СВЯЗЬ, ИНФОРМАЦИОННАЯ СИСТЕМА, БАЗА ДАННЫХ.

Учебный курсовой проект выполняется по дисциплинам «Проектирование и архитектура ПС», «Конструирование» и посвящен разработке проекта системы оператор сотовой связи. Пояснительная записка содержит следующие документы:

1. Задание.
2. Оператор сотовой связи. Техническое задание.
3. Оператор сотовой связи. Технорабочий проект.

Задание выдал:

доц. каф. ПОКС \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Игнатьев Е.Б.

Задание приняла к исполнению:

студентка гр. 3-42 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Романова Е.А.

Оглавление

[Термины, определения и сокращения 2](#_Toc122368990)

[Задание 3](#_Toc122368991)

[Введение 8](#_Toc122368992)

[Оператор сотовой связи. Техническое задание 9](#_Toc122368993)

[1 Общие сведения 9](#_Toc122368994)

[1.1 Наименование системы 9](#_Toc122368995)

[1.2 Заказчик и Разработчик системы 9](#_Toc122368996)

[1.3 Основание для разработки 9](#_Toc122368997)

[1.4 Плановые сроки начала и окончания работы по разработке проекта 9](#_Toc122368998)

[1.5 Сведения об источниках и порядке финансирования работ 9](#_Toc122368999)

[1.6 Порядок оформления и предъявления Заказчику результатов работ 9](#_Toc122369000)

[2 Назначение и цели создания системы 9](#_Toc122369001)

[2.1 Назначение системы 9](#_Toc122369002)

[2.2 Цели создания системы 10](#_Toc122369003)

[3 Характеристика объектов автоматизации 10](#_Toc122369004)

[3.1 Объекты автоматизации 10](#_Toc122369005)

[3.2 Концептуальная модель предметной области 10](#_Toc122369006)

[4 Требования к системе 12](#_Toc122369007)

[4.1 Требования к системе в целом 12](#_Toc122369008)

[4.1.1 Требования к структуре и функционированию системы 12](#_Toc122369009)

[4.1.1 Требования к численности и квалификации персонала системы и режиму его работы 13](#_Toc122369010)

[4.1.2 Показатели назначения 13](#_Toc122369011)

[4.1.3 Требования к надежности 13](#_Toc122369012)

[4.1.4 Требования безопасности 14](#_Toc122369013)

[4.1.5 Требования к эргономике и технической эстетике 14](#_Toc122369014)

[4.1.6 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы 15](#_Toc122369015)

[4.1.7 Требования к защите информации от несанкционированного доступа 15](#_Toc122369016)

[4.1.8 Требования по сохранности информации при авариях 15](#_Toc122369017)

[4.1.9 Требования к защите от влияния внешних воздействий 15](#_Toc122369018)

[4.1.10 Требования к патентной чистоте 16](#_Toc122369019)

[4.1.11 Требования по стандартизации и унификации 16](#_Toc122369020)

[4.1.12 Дополнительные требования 16](#_Toc122369021)

[4.2 Требования к функциям, выполняемым системой 16](#_Toc122369022)

[4.2.1 Модель вариантов использования 16](#_Toc122369023)

[4.2.2 Спецификация вариантов использования 17](#_Toc122369024)

[4.2.3 Временной регламент выполнения функций 22](#_Toc122369025)

[4.3 Требования к видам обеспечения 23](#_Toc122369026)

[4.3.1 Требования к математическому обеспечению 23](#_Toc122369027)

[4.3.2 Требования к информационному обеспечению 23](#_Toc122369028)

[4.3.3 Требования к лингвистическому обеспечению 23](#_Toc122369029)

[4.3.4 Требования к программному обеспечению 23](#_Toc122369030)

[4.3.5 Требования к техническому обеспечению 24](#_Toc122369031)

[4.3.6 Требования к метрологическому обеспечению 24](#_Toc122369032)

[4.3.7 Требования к организационному обеспечению 24](#_Toc122369033)

[4.3.8 Требования к методическому обеспечению 25](#_Toc122369034)

[5 Состав и содержание работ по созданию системы 25](#_Toc122369035)

[6 Порядок контроля и приемки системы 25](#_Toc122369036)

[7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие 26](#_Toc122369037)

[8 Требования к документированию 26](#_Toc122369038)

[8.1 Требования к составу документов 26](#_Toc122369039)

[8.2 Требования к оформлению документов 27](#_Toc122369040)

[8.2.1 Технорабочий проект 27](#_Toc122369041)

[9 Источники разработки 27](#_Toc122369042)

[Технорабочий проект 29](#_Toc122369043)

[1 Общие положения 29](#_Toc122369044)

[1.1 Наименование системы 29](#_Toc122369045)

[1.2 Основание для разработки 29](#_Toc122369046)

[1.3 Перечень организаций, участвующих в разработке системы, сроки выполнения этапов 29](#_Toc122369047)

[1.4 Цели, назначение и области использования 29](#_Toc122369048)

[1.5 Подтверждение соответствия проектных решений действующим нормам и правилам техники безопасности, пожаро- и взрывобезопасности 30](#_Toc122369049)

[1.6 Нормативно-технические документы, использованные при проектировании 30](#_Toc122369050)

[2 Описание процесса деятельности 30](#_Toc122369051)

[3 Основные технические решения 30](#_Toc122369052)

[3.1 Архитектура системы 30](#_Toc122369053)

[3.2 Решения по взаимосвязям со смежными системами 32](#_Toc122369054)

[3.3 Решения по режимам функционирования, диагностированию работы системы 32](#_Toc122369055)

[3.4 Решения по численности, квалификации и функциям персонала, режимам его работы, порядку взаимодействия 32](#_Toc122369056)

[3.5 Сведения об обеспечении заданных в ТЗ потребительских характеристик системы, определяющих ее качество 32](#_Toc122369057)

[3.5.1 Временные характеристики 32](#_Toc122369058)

[3.5.2 Надёжность 32](#_Toc122369059)

[3.5.3 Эргономика и техническая эстетика 32](#_Toc122369060)

[3.5.4 Защищенность информации 33](#_Toc122369061)

[3.6 Состав функций, реализуемых системой 33](#_Toc122369062)

[3.7 Решения по комплексу технических средств 33](#_Toc122369063)

[3.8 Решения по информационному обеспечению 34](#_Toc122369064)

[3.8.1 Физическая модель БД 34](#_Toc122369065)

[3.8.4. Скрипты по созданию таблиц 36](#_Toc122369066)

[3.9. Решения по программному обеспечению 39](#_Toc122369067)

[3.9.1 Классы 39](#_Toc122369068)

[3.9.1.1 BLL. Interfaces.IDbCrud 39](#_Toc122369069)

[3.9.1.2 BLL. Interfaces.IReportService 40](#_Toc122369070)

[3.9.1.3. BLL.Models.Call 40](#_Toc122369071)

[3.9.1.4. BLL.Models.Dogovor 40](#_Toc122369072)

[public DogovorModel - класс договоров. 40](#_Toc122369073)

[public DateTime Дата\_заключения – дата заключения договора. 40](#_Toc122369074)

[3.9.1.5. BLL.Models.Client 40](#_Toc122369075)

[3.9.1.6. BLL.Models.Тариф 41](#_Toc122369076)

[3.9.1.7. BLL.Services.ReportServices 41](#_Toc122369077)

[3.9.1.8. BLL.DBDataOperation 41](#_Toc122369078)

[3.9.2.0 DAL.Interfaces.IdbRepos 42](#_Toc122369079)

[3.9.2.1 DAL.Interfaces.IReportsRepository 42](#_Toc122369080)

[3.9.2.2 DAL.Interfaces.IRepository 42](#_Toc122369081)

[3.9.2.3 DAL.Models.Call 42](#_Toc122369082)

[3.9.2.4 DAL.Models.Dogovor 42](#_Toc122369083)

[3.9.2.5 DAL.Models.Client 43](#_Toc122369084)

[3.9.2.6 DAL.Models.Тариф 43](#_Toc122369085)

[3.9.2.7 DAL.Models.Тип\_звонка 43](#_Toc122369086)

[3.9.2.8 DAL.Models.Тип\_тарифа 43](#_Toc122369087)

[3.9.2.9 DAL.Repository.DbReposSQL 43](#_Toc122369088)

[3.9.2.10 DAL.Repository.DogovorRepository 44](#_Toc122369089)

[3.9.2.11 DAL.Repository.ReportRepository 44](#_Toc122369090)

[3.9.2.12 DAL.Repository.КлиентRepository 44](#_Toc122369091)

[3.9.2.13 DAL.Repository.ТарифRepository 44](#_Toc122369092)

[3.9.3.0 Wpf.Commands.RelayCommand 45](#_Toc122369093)

[3.9.3.1 Wpf.ModelView.MainFormViewModel 45](#_Toc122369094)

[3.9.3.2 Wpf.ModelView.Window1ViewModel 45](#_Toc122369095)

[3.9.3.3 Wpf.ModelView.Window2ViewModel 45](#_Toc122369096)

[3.9.3.4 Wpf.ModelView.Window3ViewModel 45](#_Toc122369097)

[3.9.3.5 Wpf.ModelView.Window6ViewModel 46](#_Toc122369098)

[3.9.3.6 Wpf.ModelView.Util.NinjectRegistrations 46](#_Toc122369099)

[3.9.4 Диаграмма последовательности для варианта «Добавить клиента» 46](#_Toc122369100)

[3.9.5 Диаграмма коммуникации для варианта «Добавить клиента» 47](#_Toc122369101)

[3.9.6 Диаграмма классов для варианта «Добавить клиента» 47](#_Toc122369102)

[3.9.7 Диаграмма состояний для метода «Архивация тарифа» 48](#_Toc122369103)

[3.9.8 Диаграмма деятельности для варианта «Удалить клиента» 49](#_Toc122369104)

[4 Мероприятия по подготовке к вводу системы в действие 49](#_Toc122369105)

[4.1 Мероприятия по приведению информации к виду, пригодному для обработки на ЭВМ 49](#_Toc122369106)

[4.2 Мероприятия по обучению и проверке квалификации персонала 50](#_Toc122369107)

[4.3 Мероприятия по созданию необходимых подразделений и рабочих мест 50](#_Toc122369108)

[4.4 Мероприятия по изменению объекта автоматизации 50](#_Toc122369109)

[5 Источники разработки 50](#_Toc122369110)

# Введение

Оператору сотовой связи нужно работать с базой клиентов, изменять их данные, обновлять, добавлять новых клиентов, предоставлять данные о звонках.

Перед руководством организации встали задачи улучшения работы с базами данных клиентов, автоматизация этого процесса, унифицировать этот процесс, собрав все функции работы с базой в одну программу.

Поэтому, создание системы, автоматизирующей работу с базой данных клиентов оператора сотовой связи, является актуальной задачей.

Приложение будет иметь вид толстого клиента, которое будет общаться напрямую с базой данных. Оператор будет работать с данными базы клиентов: добавлять, удалять клиентов, редактировать их данные, формировать отчеты по звонкам определенного клиента, добавлять, изменять тарифные планы.

# Оператор сотовой связи. Техническое задание

# Общие сведения

## Наименование системы

Полное наименование системы — «Оператор сотовой связи».

Условное обозначение системы — «MO» (Mobile Operator).

## Заказчик и Разработчик системы

Заказчик системы: Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Ивановский государственный энергетический университет имени В.И. Ленина» (ИГЭУ); 153003, г. Иваново, ул. Рабфаковская, д. 34.

Разработчик системы: Романова Екатерина Алексеевна, студентка группы 3-42.

## Основание для разработки

Разработка ведется на основании задания на курсовую работу по дисциплине «Проектирование программного обеспечения».

Задание утверждено на заседании кафедры ПОКС 26.08.2022 и выдано преподавателем кафедры Игнатьевым Е.Б.

## Плановые сроки начала и окончания работы по разработке проекта

Начало: 23 сентября 2022 г.

Окончание: 25 декабря 2022 г.

## Сведения об источниках и порядке финансирования работ

Финансирование работ отсутствует.

## Порядок оформления и предъявления Заказчику результатов работ

Разработчик оформляет результаты работ над проектом в два этапа, в виде эскизного проекта и технического проекта; и передает их Заказчику.

# Назначение и цели создания системы

## Назначение системы

MO-система предназначена для работы с базой клиентов оператора сотовой связи.

## Цели создания системы

Основными целями создания системы являются:

- объединение функционала работы с базой клиентов в одну программу

- обеспечение прозрачности взаиморасчетов между сотовым оператором и клиентом.

# Характеристика объектов автоматизации

## Объекты автоматизации

Объектами автоматизации являются процессы ведения договоров с клиентами и контроль оплаты тарифов.

Ведением базы занимается оператор. При поступлении информации, база обновляется оператором (добавление нового клиента, добавление и редактирование тарифов, изменение данных клиента, удаление клиента, информация о его входящих или исходящих звонков).

У каждого пользователя свой набор данных.

Звонки бывают–междугородние и международные.

## Концептуальная модель предметной области

В результате обследования предметной области была разработана модель объектов предметной области (Domain Object Model – DOM), описывающая классы предметной об­ласти и связи между ними.

Рис. 3.1 представляет диаграмму классов для концептуальных классов, не­обходимых в итерации 1.

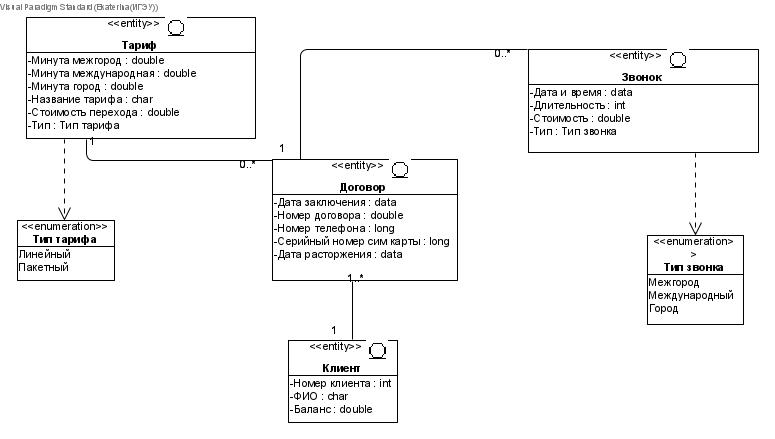


Рис. 3.1. Концептуальная модель предметной области

Диаграмма классов содержит шесть классов: Тариф, Тип\_тарифа, Клиент, Звонок, Тип\_звонка, Договор. В таблицах 3.1-3.7 перечислены свойства атрибутов.

Таблица 3.1 - Атрибуты класса «Тариф» (Tarif)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Код | Тип | Видимость | Только для чтения |
| Код\_тарифа | Tarif\_id | Int | public | false |
| Название\_тарифа | Tarif\_name | Char | public | false |
| Стоимость\_перехода | Tarif\_costcross | Money | public | false |
| Минута\_межгород | Tarif\_mingorod | Money | public | false |
| Минута\_международная | Tarif\_minworld | Money | public | false |

Таблица 3.2 - Атрибуты класса «Клиент» (User)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Код | Тип | Видимость | Только для чтения |
| Номер\_клиента | User\_id | Int | public | false |
| ФИО | User\_FIO | Char | public | false |
| Баланс | User\_Balance | Money | public | false |

Таблица 3.3 - Атрибуты класса «Звонок» (Call)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Код | Тип | Видимость | Только для чтения |
| Дата\_и\_время | Call\_date | Date | public | false |
| Длительность | Call\_time | Int | public | false |
| Стоимость | Call\_cost | Money | public | false |

Таблица 3.4 - Атрибуты класса «Тип\_звонка» (Typecall)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Код | Тип | Видимость | Только для чтения |
| Код\_типа\_звонка | Typecall\_id | Int | public | false |
| Название\_типа\_тарифа | Typecall\_type | Char | public | false |

Таблица 3.5 - Атрибуты класса «Договор» (Contract)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Код | Тип | Видимость | Только для чтения |
| Номер\_договора | Cont\_id | Int | public | false |
| Номер\_договора | Cont\_nmbr | Int | public | false |
| Дата\_заключения | Cont\_datebegin | Date | public | false |
| Дата\_расторжения | Cont\_dateend | Date | public | false |
| Серийный\_номер\_сим\_карты | Cont\_simnmbr | Char | public | false |
| Номер\_телефона | Cont\_phone | Char | public | false |

Таблица 3.6 - Атрибуты класса «Тип\_тарифа» (Typetarif)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Код | Тип | Видимость | Только для чтения |
| Код\_типа\_тарифа | Typetarif\_id | Int | public | false |
| Название\_типа\_тарифа | Typetarif\_type | Char | public | false |

Каждому пользователю соответствует один тариф, один или несколько договоров, один или более звонков, каждому звонку соответствует тип звонка(городской, междугородний, мировой).

# Требования к системе

## Требования к системе в целом

### Требования к структуре и функционированию системы

**Архитектура системы**

Система MO должна иметь двухуровневую клиент-серверную архитектуру:

На уровне хранения данных размещается сервер БД, а на прикладном уровне насыщенное клиентское приложение.

**Информационный обмен между компонентами системы**

Входящие в состав MO подсистемы в процессе функционирования должны обмениваться информацией на основе открытых форматов обмена данными по протоколам на основе TCP/IP.

**Взаимосвязи со смежными системами**

Требуется обеспечить взаимодействие с системой мобильной связи, чтобы получать данные о звонках конкретного клиента для того, чтобы иметь актуальную информацию о его счете и данных его переговорах.

**Перспективы развития, модернизации системы**

Система должна реализовывать возможность дальнейшей модернизации как программного обеспечения, так комплекса технических средств.

### Требования к численности и квалификации персонала системы и режиму его работы

Для эксплуатации МО должны быть предусмотрены следующие роли пользователей:

1. Администратор
2. Оператор

Администратор должен обладать высоким уровнем квалификации и практическим опытом выполнения работ по установке, настройке и администрированию программных и технических средств, применяемых в системе и ведением тарифов.

Оператор должен иметь опыт работы с персональным компьютером на базе операционных систем Microsoft Windows на уровне квалифицированного пользователя и свободно осуществлять базовые операции в стандартных Windows-приложениях.

Рекомендуемая численность для эксплуатации системы MO:

- Администратор – 1 штатная единица

- Оператор – число штатных единиц определяется структурой предприятия;

### Показатели назначения

Система должна предусматривать возможность масштабирования по производительности и объему обрабатываемой информации без модификации ее программного обеспечения путем модернизации используемого комплекса технических средств. Возможности масштабирования должны обеспечиваться средствами используемого базового программного обеспечения.

### Требования к надежности

Система должна быть доступна 24 часа каждый день недели. Не должно быть никакого связанного с БД времени простоя.

Отказ компонентов программного обеспечения не должен ставить под угрозу корректность и целостность БД. Пользователь должен иметь возможность повторно начать программу после отказа и найти информацию БД непротиворечивой и не повреждённой в результате отказа.

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

– при сбоях в системе электроснабжения аппаратной части, приводящих к перезагрузке ОС, восстановление программы должно происходить после перезагрузки ОС и запуска Системы;

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

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

Система должна обеспечивать корректную обработку ситуаций, вызванных недопустимыми и несогласованными значениями входных данных. В указанных случаях пользователю (оператору, администратору) должны выдаваться соответствующие уведомления, после чего система должна возвращаться в рабочее состояние.

Для обеспечения устойчивости к отказам электроснабжения все устройства хранения и обработки информации должны быть подключены к электросети через источники бесперебойного питания.

### Требования безопасности

Все технические решения, использованные при создании системы, а также при определении требований к аппаратному обеспечению, должны соответствовать действующим нормам и правилам техники безопасности, пожаробезопасности и взрывобезопасности, а также охраны окружающей среды при эксплуатации.

### Требования к эргономике и технической эстетике

Взаимодействие пользователей с прикладным программным обеспечением, входящим в состав системы должно осуществляться посредством визуального графического интерфейса (GUI). Интерфейс системы должен быть понятным и удобным, не должен быть перегружен графическими элементами и должен обеспечивать быстрое отображение экранных форм. Навигационные элементы должны быть выполнены в удобной для пользователя форме. Средства редактирования информации должны удовлетворять принятым соглашениям в части использования функциональных клавиш, режимов работы, поиска, использования оконной системы.

### Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Система должна быть рассчитана на эксплуатацию в составе программно-технического комплекса Заказчика и учитывать разделение ИТ-инфраструктуры Заказчика на внутреннюю и внешнюю. Техническая и физическая защита аппаратных компонентов системы, носителей данных, бесперебойное энергоснабжение, резервирование ресурсов, текущее обслуживание реализуется техническими и организационными средствами, предусмотренными в ИТ инфраструктуре Заказчика.

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

### Требования к защите информации от несанкционированного доступа

Необходимо обеспечивать защиту от несанкционированного доступа (НСД) на уровне не ниже установленного требованиями, предъявляемыми к категории 1Д по классификации действующего РД Гостехкомиссии России [6].

Разрабатываемая система должна обеспечивать разграничение доступа на уровне отдельных программных модулей и структур данных.

### Требования по сохранности информации при авариях

Программное обеспечение Системы должно восстанавливать свое функционирование при корректном перезапуске аппаратных средств. Должна быть предусмотрена возможность организации автоматического и (или) ручного резервного копирования данных системы средствами системного и базового программного обеспечения (ОС, СУБД), входящего в состав программно-технического комплекса Заказчика.

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

### Требования к защите от влияния внешних воздействий

Требования к защите от влияния внешних воздействий не предъявляются.

### Требования к патентной чистоте

Установка системы в целом, как и установка отдельных частей системы не должна предъявлять дополнительных требований к покупке лицензий на программное обеспечение сторонних производителей, кроме программного обеспечения, указанного в разделе 4.3.4.

### Требования по стандартизации и унификации

Требования не предъявляются.

### Дополнительные требования

Специальные требования не предъявляются.

## Требования к функциям, выполняемым системой

Приложение будет иметь вид толстого клиента, которое будет общаться напрямую с базой данных. Оператор будет работать с данными базы клиентов: добавлять, удалять клиентов, редактировать их данные, формировать отчеты по звонкам определенного клиента, вести статистику баланса пользователя, добавлять, изменять и удалять тарифные планы.

### Модель вариантов использования

Рис. 4.1 представляет диаграмму вариантов использования, предназначенную для представления цели, намеченных функциональных возможностей и пред­положений

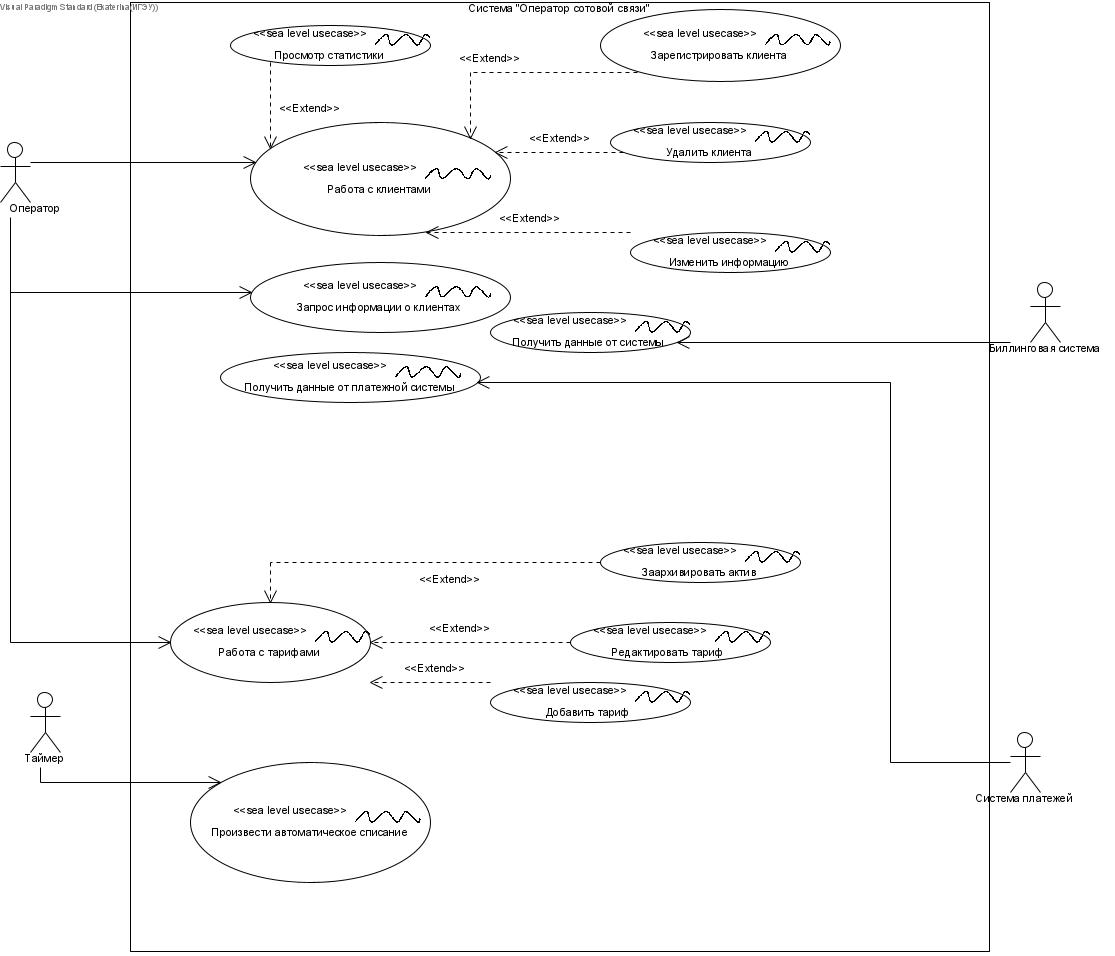


Рис. 4.1. Диаграмма вариантов использования

Актером системы является Оператор. Оператор может добавлять клиента в базу, просмотреть всех клиентов из базы с подробной информацией о каждом. Также оператор может работать с отдельным клиентом: просмотреть статистику его звонков, изменить информацию о клиенте или удалить клиента. Он просматривает все доступные тарифы и получает детальную информацию о них, может работать с ними - добавлять новый или редактировать старые.

### Спецификация вариантов использования

#### 1)Название ВИ: Запросить информации о тарифах

**Краткое описание.** В систему отправляется запрос на просмотр информации о тарифах, система формирует таблицу с данными по тарифам

**Участвующие актеры:**Оператор

**Поток событий**

**Основной поток**

1. Оператор вводит запрос на получение информации о тарифах с параметрами.
2. Система находит информацию в БД
3. Система формирует расширенную таблицу с выбранными тарифами

**Альтернативные потоки**

**1А.** Параметры введены некорректно

**1А1.** Система выводит сообщение об ошибке и переходит в главное меню

**Предусловия**

Оператор идентифицирован в системе

**Постусловие:**

Таблица с данными по тарифам

#### Название ВИ: Запросить информации о клиентах

**Краткое описание:** В систему отправляется запрос на просмотр информации о клиентах, система формирует таблицу с данными по клиентам

**Участвующие актеры:** Оператор

**Поток событий**

**Основной поток**

1. Оператор вводит запрос на получение информации о клиентах с параметрами(клиенты с нулевым\ненулевым балансом, с подключенным определенным тарифом )
2. Система находит информацию в БД
3. Система формирует расширенную таблицу с выбранными клиентами

**Альтернативные потоки**

**1А.** Параметры введены некорректно

**1А1.** Система выводит сообщение об ошибке и переходит в главное меню

**Предусловие:**

Оператор идентифицирован в системе

**Постусловие:**

Таблица с полученными данными о клиентах

#### Название ВИ: Добавить клиента в базу

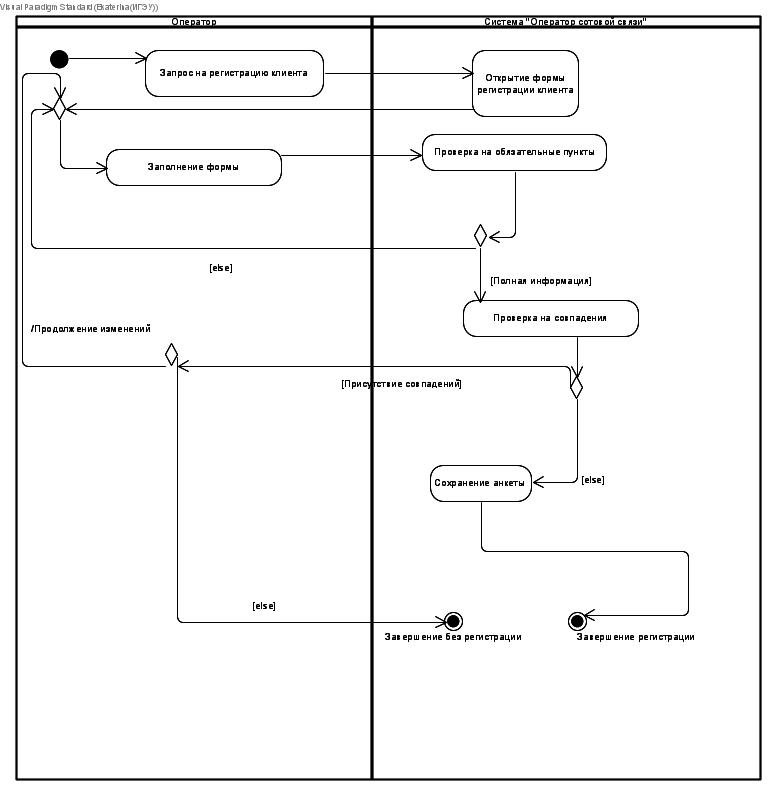


Рис. 4.2. сло для ВИ «Зарегистрировать клиента»

**Краткое описание.** Оператор заполняет данные по клиенту и информация отправляется в базу

**Участвующие актеры:** Оператор

**Поток событий**

**Основной поток**

1. Оператор вводит данные о клиенте. Оператору предлагается ввести дополнительные параметры
2. Система отправляет данные о новом клиенте в базу для проверки
3. Система добавляет клиента и выводит сообщение об успехе

**Альтернативные потоки**

**2А.** Один из уникальных параметров совпадает у нового пользователя и одного из базы

**2А1.** Система выводит сообщение об ошибке и переходит в главное меню

**Предусловие:**

Оператор идентифицирован в системе

**Постусловие:**

Добавленный в базу новый клиент.

#### Название ВИ: Просмотреть статистику клиента

**Краткое описание:** Оператор отправляет запрос на получение статистики клиента, система выводит данные по пользователю в виде таблицы

**Участвующие актеры:** Оператор

**Поток событий**

**Основной поток**

1. Оператор вводит запрос на определенного клиента
2. Система находит клиента и выводит детальную информацию обо всех вызовах клиента в виде расширенной таблицы

**Альтернативные потоки**

**1А.** Клиента с таким номером нет в базе

**1А1.** Система выводит сообщение об ошибке и переходит в главное меню

**Предусловие:**

Оператор идентифицирован в системе

**Постусловие:**

Статистика клиента

#### Название ВИ: Изменить информацию о клиенте

**Краткое описание.** Оператор вводит запрос на изменение информации по определенному пользователю, система обновляет данные базы

**Участвующие актеры.** Оператор

**Поток событий**

**Основной поток**

1. Оператор вводит запрос на определенного клиента по его коду
2. Система находит пользователя
3. Оператор вводит новые данные для пользователя
4. Система сохраняет их в базе

**Альтернативные потоки**

**2А.**Клиента с таким номером нет в базе

**2А1.** Система выводит сообщение об ошибке и переходит в главное меню

**4А.** Данные были введены некорректно

**4А1.** Система выводит сообщение об ошибке и переходит в главное меню

**Предусловие:**

Оператор идентифицирован в системе

**Постусловие:**

Обновленная база данных

#### Название ВИ: Удалить клиента

**Краткое описание.** Оператор вводит запрос на удаление определенного клиента из базы по его номеру

**Участвующие актеры.** Оператор

**Поток событий**

**Основной поток**

1. Оператор вводит запрос на определенного клиента по его номеру
2. Система находит пользователя
3. Оператор удаляет пользователя
4. Система удаляет все данные пользователя во всех таблицах

**Альтернативные потоки**

**2А.** Клиента с таким номером нет в базе

**2А1.** Система выводит сообщение об ошибке и переходит в главное меню

**Предусловие:**

Оператор идентифицирован в системе

**Постусловие:**

Уменьшенная база данных

#### Название ВИ: Добавить тариф

**Краткое описание:** Оператор заполняет данные нового тарифа и информация отправляется в базу

**Участвующие актеры:** Оператор

**Поток событий**

**Основной поток**

1. Оператор вводит данные тарифа.
2. Система отправляет данные о новом тарифе в базу для проверки
3. Система добавляет тариф и выводит сообщение об успехе

**Альтернативные потоки**

**2А.** Один из уникальных параметров совпадает у нового тарифа и одного из базы

**2А1.** Система выводит сообщение об ошибке и переходит в главное меню

**Предусловия**

Оператор идентифицирован в системе

**Предусловие:**

Новый тариф

#### Название ВИ: Редактировать тариф

**Краткое описание.** Оператор вводит запрос на изменение информации одного из тарифов, система обновляет данные базы

**Участвующие актеры.** Оператор

**Поток событий**

**Основной поток**

1. Оператор выбирает тариф, который он хочет редактировать
2. Оператор вводит новые данные для тарифа
3. Система сохраняет их в базе

**Альтернативные потоки**

**3А.** Данные были введены некорректно

**3А1.** Система выводит сообщение об ошибке и переходит в главное меню

**Предусловие:**

Оператор идентифицирован в системе

**Постусловие:**

Обновленный тариф, новые условия

### Временной регламент выполнения функций

Время отклика для функций просмотра данных по тарифам и клиентам, добавление клиента в базу, удаление клиента из базы, просмотр статистики должно быть меньше 5 секунд с 90‑процентной вероятностью.

Время отклика для функции изменение информации, добавление и изменение тарифа должно быть меньше 2 секунд с 90-процентной вероятностью.

## Требования к видам обеспечения

### Требования к математическому обеспечению

Требования к математическому обеспечению не предъявляются.

### Требования к информационному обеспечению

Уровень хранения данных в Системе должен быть построен на платформе реляционной СУБД. Для обеспечения целостности данных должны использоваться встроенные механизмы СУБД.

База данных предназначена для хранения:

* Данных о клиентах
* Данных о тарифах
* Данных о договорах

### Требования к лингвистическому обеспечению

Программное обеспечение системы должно быть разработано на языке программирования c#.

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

### Требования к программному обеспечению

Проект должен использовать СУБД SQL, но он должен быть легко перестраиваемым для других реляционных БД.

Для создания кода должна использоваться управляемая тестированием разработка. Для проверки кода - приемочные испытания. Тестируемые единицы, полученные в результате управляемой тестированием разработ­ки и приемочных испытаний, используются для регрессионного тестиро­вания,когда код будет заменен на итерацию 2.

Разрабатываемая Система должна быть рассчитана на функционирование в следующей программной среде:

**Серверная группа**

ПО, устанавливаемое на компьютеры серверной группы:

1. Базовая ОС – Microsoft Windows 2019 Server.
2. Система управления базами данных – Microsoft SQL Server 2019.
3. Firewall для защиты внутренних ресурсов системы, при наличии подключения к транзитным провайдерам услуг передачи данных – Microsoft ISA Server.

**Рабочие станции**

Типовое программное обеспечение, устанавливаемое на рабочие станции:

1. Базовая операционная система: Windows 10 Pro.

### Требования к техническому обеспечению

Техническое обеспечение системы должно максимально и наиболее эффективным образом использовать существующие технические средства.

В состав комплекса должны входить следующие технические средства:

1) сервер БД;

2) персональные компьютеры (ПК) пользователей.

Минимальные требования к характеристикам компонентов технического обеспечения, при которых значения временных параметров Системы должны соответствовать предъявленным в ТЗ требованиям (п. 4.2.3):

1. для сервера БД:

– процессор – Intel Xeon E-2126G;

– объем оперативной памяти – 64 Гб;

– дисковая подсистема – 8 х 146 Гб;

– устройство чтения компакт-дисков (DVD-ROM);

– сетевой адаптер – 500 Мбит/с.

1. для ПК пользователя:

– процессор – Intel Core i7;

– объем оперативной памяти – 16 Гб;

– дисковая память – 512 Гб;

– сетевой адаптер – 500 Мбит/с.

### Требования к метрологическому обеспечению

Требования к метрологическому обеспечению не предъявляются.

### Требования к организационному обеспечению

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

Заказчиком должны быть подготовлены изменения в положения о структурных подразделениях, в которых будет эксплуатироваться система МО.

Заказчиком должны быть подготовлены изменения к действующим должностным инструкциям, для персонала, который будет участвовать в эксплуатации Системы.

Должностные инструкции должны определять функциональные обязанности и ответственность сотрудников, участвующих в обслуживании и эксплуатации Системы:

В обязанности Оператора должны войти:

- изменение информации клиентам

-добавление клиентов

- сбор статистических данных:

- изменение информации о тарифах, удаление и создание новых

- удаление клиентов

- заключение, изменение, удаление договоров

Заказчиком должен быть подготовлен приказ о приёмке системы МО и вводе её в эксплуатацию с указанием ответственных за эксплуатацию системы.

### Требования к методическому обеспечению

Требования к методическому обеспечению не предъявляются.

# Состав и содержание работ по созданию системы

Разработка и сдача проекта должна вестись по этапам (Таблица 5.1).

Таблица 5.1. Этапы работ над проектом

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование этапа** | **Результаты этапа** | **Дата начала этапа** | **Дата заверше­ния этапа** |
| 1. Разработка, согласование и утверждение технического задания | Техническое задание | 23.09.2022 | 5.10.2022 |
| 2. Разработка, согласование и утверждение эскизного проекта | Эскизный проект | 7.10.2022 | 25.10.2022 |
| 3. Разработка, согласование и утверждение технического проекта | Технический проект | 28.10.2022 | 13.12.2022 |
| 4. Подготовка презентации и защита проекта | Пояснительная записка и презентация | 16.12.2022 | 25.12.2022 |

# Порядок контроля и приемки системы

По окончании работ проект принимается Приёмной комиссией.

Заседание комиссии проводится в конце 5-го семестра перед зачётной неделей.

Проект подлежит защите. Защита проекта проводится Разработчиком перед членами Приёмной комиссии.

Приёмная комиссия назначается из числа преподавателей кафедры ПОКС.

Комиссии предоставляется полностью оформленная и подписанная пояснительная записка и презентация.

Проект принимается, если он удовлетворяет всем пунктам данного технического задания. По результатам защиты проекта Приёмная комиссия выставляет оценку.

# Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

Для ввода Системы в действие необходимо:

1. подготовить у Заказчика всё необходимое техническое обеспечение;
2. установить на сервер и клиентские ПК системное, базовое и прикладное ПО;
3. установить на сервер БД;
4. ввести данные в справочники БД

- сведения о доступных тарифах;

- сведения о заключенных договорах;

- сведения о клиентах;

1. подготовить организационное обеспечение;
2. провести обучение Операторов и Администраторов;
3. провести испытания системы.

# Требования к документированию

## Требования к составу документов

По окончании работ над проектом все разработанные документы объединяются в Пояснительную записку. Она должна содержать:

1. Титульный лист,
2. Аннотацию,
3. Оглавление,
4. Термины, определения и сокращения, использованные в Пояснительной записке,
5. Задание,
6. Введение,
7. Техническое задание,
8. Технорабочий проект,
9. Список литературы.

Заказчику предоставляются:

1. Пояснительная записка в формате MS Word 2010;
2. Пояснительная записка, распечатанная на бумаге формата А4 – 1 экземпляр;
3. Презентация (в формате MS PowerPoint), демонстрирующая основные проектные решения.

## Требования к оформлению документов

Техническое задание, Технорабочий проект и Технический проект оформляются в соответствии с ГОСТ 34.201-89 [2], ГОСТ 34.602-89 [4] и РД 50-34.698.90 [5].

### Технорабочий проект

Технорабочий проект должен содержать следующие разделы:

1. общие положения;
2. описание процесса деятельности;
3. основные технические решения;
4. мероприятия по подготовке объекта автоматизации к вводу системы в действие;
5. схема функциональной структуры;
6. описание автоматизируемых функций;
7. описание комплекса технических средств;
8. описание программного обеспечения;
9. описание информационного обеспечения;
10. описание организационной структуры;
11. руководство пользователя.

# Источники разработки

При разработке ТЗ использовались следующие источники:

1. ГОСТ 34.003-90. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения [1].
2. ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем [2].
3. ГОСТ 34.601-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания [3].
4. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы [4].
5. РД 50-34.698.90. Методические указания. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов [5].
6. РД Гостехкомиссии России. Безопасность информационных технологий. Критерии оценки безопасности информационных технологий.- 2002 г. [6].
7. Практическая программная инженерия на основе учебного примера / Л.А. Мацяшек, Б.Л. Лионг. – М.: БИНОМ. Лаборатория знаний, 2009. – 956 с. [7].

# Технорабочий проект

# Общие положения

## Наименование системы

Полное наименование системы — «Оператор сотовой связи».

Условное обозначение системы — «MO» (Mobile Operator).

## Основание для разработки

Разработка ведется на основании задания на курсовой проект по дисциплинам: «Проектирование и архитектура ПС», «Проектирование человеко-машинного интерфейса», «Конструирование».

Задание утверждено на заседании кафедры ПОКС 01.09.2022 и выдано преподавателем кафедры Игнатьевым Е.Б.

## Перечень организаций, участвующих в разработке системы, сроки выполнения этапов

Заказчик системы: Федеральное государственное бюджетное образовательное учреждение высшего образования «Ивановский государственный энергетический университет имени В.И. Ленина» (ИГЭУ); 153003, г. Иваново, ул. Рабфаковская, д. 34.

Разработчик системы: Романова Екатерина Алексеевна, студентка 3-42.

Сроки выполнения этапов работы над проектом показаны в таблице 1.1.

**Таблица 1.1. Этапы работ над проектом**

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование этапа** | **Результаты этапа** | **Дата начала этапа** | **Дата завершения этапа** |
| 1. Разработка, согласование и утверждение технорабочего проекта | Технорабочий проект | 28.10.2022 | 13.12.2022 |
| 2. Подготовка презентации и защита проекта | Пояснительная записка и презентация | 16.12.2022 | 25.12.2022 |

## Цели, назначение и области использования

Основными целями создания системы являются:

- детализированное отслеживание подключаемых тарифов и зарегистрированных клиентов, заключенных договоров

- формирование отчетов

MO-система предназначена для управления тарифами, клиентами и договорами.

## Подтверждение соответствия проектных решений действующим нормам и правилам техники безопасности, пожаро- и взрывобезопасности

Все технические решения, использованные при разработке проекта, соответствуют действующим нормам и правилам техники безопасности, пожаробезопасности и взрывобезопасности, а также охраны окружающей среды при эксплуатации.

## Нормативно-технические документы, использованные при проектировании

При проектировании использовалась спецификация текущей версии UML - Documents associated with. Unified Modeling Language (UML), v2.4.1. Release date: August 2011.- http://www.omg.org/spec/UML/2.4.1/ [8].

# Описание процесса деятельности

Система MO автоматизирует управлением и добавлением новых тарифов и подключением новых клиентов.

Оператор является служащим отдела по работе с клиентами, тарифами и договорами:

- заархивировать тариф;

- изменить информацию тарифа, что включает в себя: повышение/понижение стоимости минут и перехода, изменение наименования и типа;

- добавить новый тариф;

- заключать договора с новыми клиентами;

- изменять информацию о клиентах.

# Основные технические решения

## Архитектура системы

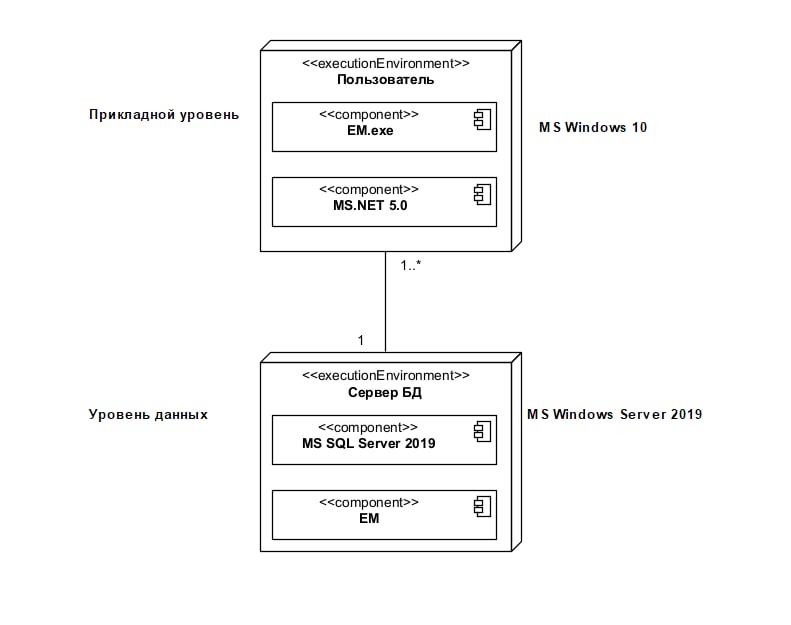
Система MO имеет двухуровневую клиент-серверную архитектуру (Рис. 3.1).

На уровне данных размещается подсистема хранения данных. Она содержит СУБД и БД.

На прикладном уровне размещается подсистема приложений. Она содержит байт-код приложения.

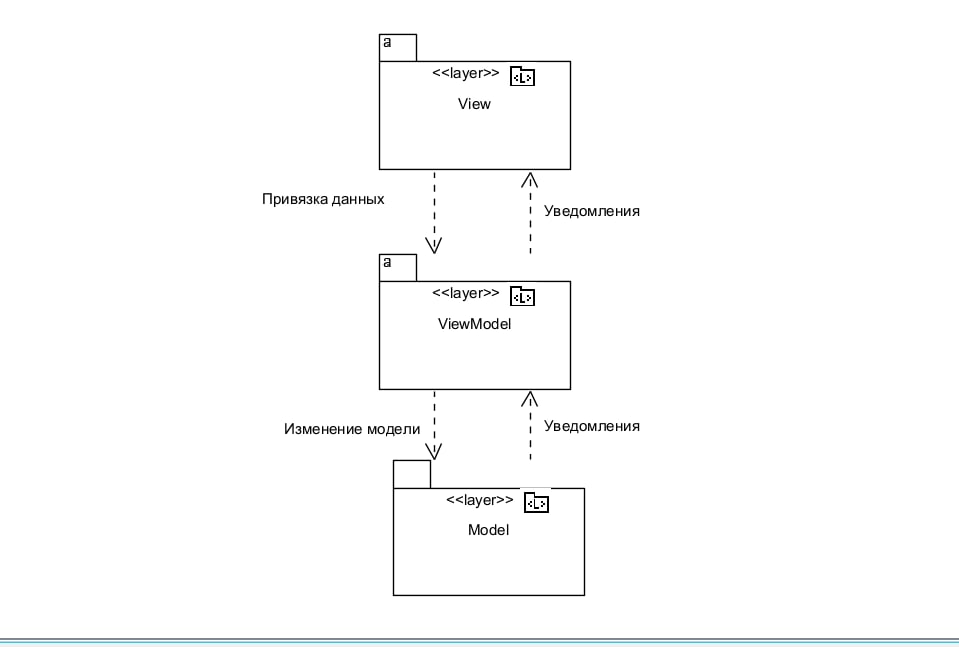
Подсистема хранения данных предназначена для хранения оперативных данных системы.

Подсистема приложений предназначена для автоматизации процедур управления системой тарифов и клиентов.



**Рис. 3.1. Диаграмма развёртывания**

Функциональное деление компонентов программного обеспечения по слоям выполняется в соответствии со структурным паттерном MVVM (Рис. 3.2).



**Рис. 3.2. Структурный шаблон MVVM**

## Решения по взаимосвязям со смежными системами

Система взаимодействует с биллинговой системой, предоставляющей данные о совершенных пользователями звонках.

## Решения по режимам функционирования, диагностированию работы системы

В работе системы можно выделить следующие режимы работы:

1. Работа авторизованного пользователя

Специальных процедур диагностирования состояния компонентов системы не предусмотрено.

## Решения по численности, квалификации и функциям персонала, режимам его работы, порядку взаимодействия

MO — многопользовательское приложение.

Авторизация пользователя и его права доступа к различным средствам приложения управляются централизованно из БД, которая соединена с прикладной программой.

Не требуется никакого обучения для пользователя, знакомого с компьютером, чтобы он мог использовать 1-ю версию системы. Простого объяснения цели и основных особенностей приложения будет достаточно, чтобы использовать программу.

## Сведения об обеспечении заданных в ТЗ потребительских характеристик системы, определяющих ее качество

### Временные характеристики

Время отклика для всех подпотоков с 90-процентной вероятностью будет меньше 5 секунд за счет использования высокопроизводительной СУБД - (Microsoft SQL Server 2019), которая обеспечивает одновременную работу 32-х потоков.

### Надёжность

Для обеспечения устойчивости к отказам электроснабжения все устройства хранения и обработки информации подключаются к электросети через источники бесперебойного питания соответствующей мощности.

### Эргономика и техническая эстетика

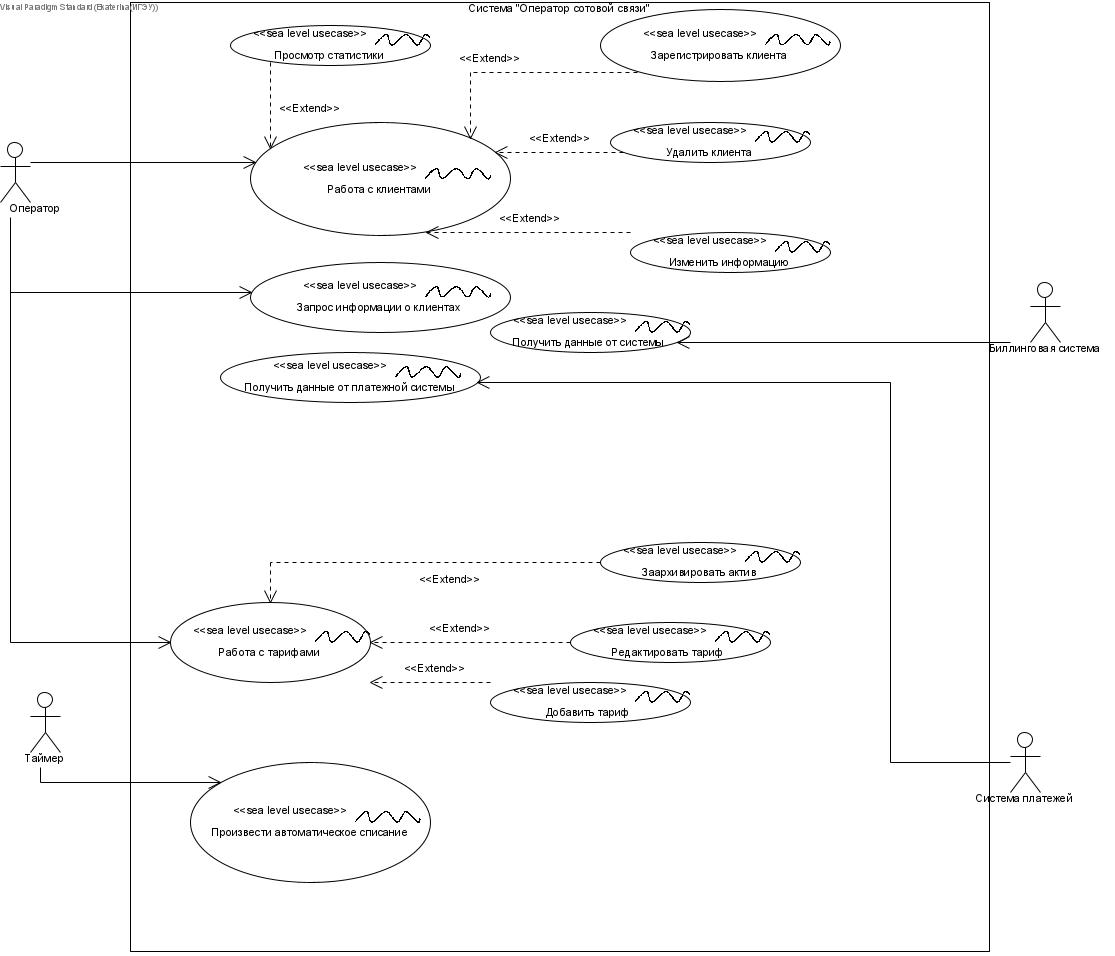
Требования эргономичности и технической эстетики начиная со 2-ой версии будут обеспечены использованием набора компонентов.

### Защищенность информации

Для восстановления работоспособности системы после сбоев предусмотрено резервное копирование БД средствами СУБД SQL.

## Состав функций, реализуемых системой

На диаграмме вариантов использования (Рис. 3.3) приведены функции системы.



**Рис. 3.3. Диаграмма вариантов использования**

Версия 1 системы рассматривает следующие варианты использования для Оператора:

* + - 1. Работа с клиентами
      2. Работа с тарифами
      3. Работа с договорами

## Решения по комплексу технических средств

В соответствии с ТЗ:

1. в качестве сервера БД будет использован компьютер, установленный в специальном серверном помещении;
2. в качестве ПК пользователей будут использованы компьютеры установленные в настоящее время на рабочих местах сотрудников;
3. все компьютеры объединены локальной вычислительной сетью с пропускной способностью 100 Мбит/с.

**Характеристики сервера:**

– процессор – Intel Xeon E-2126G;

– объем оперативной памяти – 64 Гб;

– дисковая подсистема – 8 х 146 Гб;

– устройство чтения компакт-дисков (DVD-ROM);

– сетевой адаптер – 500 Мбит/с.

**Минимальные характеристики для ПК пользователя**

– процессор – Intel Core i7;

– объем оперативной памяти – 16 Гб;

– дисковая память – 512 Гб;

– сетевой адаптер – 500 Мбит/с.

## Решения по информационному обеспечению

Компонентом информационного обеспечения является системная БД «MO».

Версия 1 MO предполагает, что БД существует заранее и загружена данными относительно доступных тарифов, их типов и типов звонков.

### Физическая модель БД

Рис. 3.5 представляет физическую модель БД.



**Рис. 3.5. Физическая модель БД MO**

Модель определяет физические характеристики (индексы) и использует типы данных конкретной СУБД – Microsoft SQL Server.

Таблица 3.1 - Атрибуты класса «Тариф» (Tarif)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Код | Тип | Видимость | Только для чтения |
| Код\_тарифа | Tarif\_id | Int | public | false |
| Название\_тарифа | Tarif\_name | Varchar(50) | public | false |
| Стоимость\_перехода | Tarif\_costcross | Money | public | false |
| Минута\_межгород | Tarif\_mingorod | Money | public | false |
| Минута\_международная | Tarif\_minworld | Money | public | False |
| Код\_типа\_тарифаFK | Tarif\_tip | Int | public | false |

Таблица 3.2 - Атрибуты класса «Клиент» (Client)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Код | Тип | Видимость | Только для чтения |
| Номер\_клиента | Client\_id | Int | public | false |
| ФИО | Client\_FIO | Varchar(50) | public | false |
| Баланс | Client\_Balance | Money | public | false |

Таблица 3.3 - Атрибуты класса «Звонок» (Call)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Код | Тип | Видимость | Только для чтения |
| Дата\_и\_время | Call\_date | Date | public | false |
| Длительность | Call\_time | Int | public | false |
| Стоимость | Call\_cost | Money | public | false |
| Номер\_договораFK | Call\_dogovor | Int | public | false |
| Код\_типа\_звонкаFK | Call\_tip | Int | public | false |

Таблица 3.4 - Атрибуты класса «Тип\_звонка» (Typecall)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Код | Тип | Видимость | Только для чтения |
| Код\_типа\_звонка | Typecall\_id | Int | public | false |
| Название\_типа\_тарифа | Typecall\_type | Varchar(50) | public | false |

Таблица 3.5 - Атрибуты класса «Договор» (Contract)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Код | Тип | Видимость | Только для чтения |
| Номер\_договора | Cont\_id | Int | public | false |
| Номер\_договора | Cont\_nmbr | Int | public | false |
| Дата\_заключения | Cont\_datebegin | Date | public | false |
| Дата\_расторжения | Cont\_dateend | Date | public | false |
| Серийный\_номер\_сим\_карты | Cont\_simnmbr | Varchar(50) | public | false |
| Номер\_телефона | Cont\_phone | Varchar(50) | public | false |
| Код\_тарифаFK | Cont\_tarif | Int | public | false |
| Номер\_клиентаFK | Cont\_Client | Int | public | false |

Таблица 3.6 - Атрибуты класса «Тип\_тарифа» (Typetarif)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Код | Тип | Видимость | Только для чтения |
| Код\_типа\_тарифа | Typetarif\_id | Int | public | false |
| Название\_типа\_тарифа | Typetarif\_type | Char | public | false |

### Скрипты по созданию таблиц

1. **Договор**

USE [operator]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Dogovor] Script Date: 17.11.2022 8:29:42 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Dogovor](

[Дата заключения] [date] NOT NULL,

[Номер договора] [int] NOT NULL,

[Номер телефона] [nvarchar](50) NOT NULL,

[Серийный номер сим карты] [nvarchar](50) NOT NULL,

[Дата расторжения] [date] NULL,

[Код тарифа FK] [int] NOT NULL,

[Номер клиента FK] [int] NOT NULL,

CONSTRAINT [PK\_Договор] PRIMARY KEY CLUSTERED

(

[Номер договора] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Dogovor] WITH CHECK ADD CONSTRAINT [FK\_Договор\_Клиент] FOREIGN KEY([Номер клиента FK])

REFERENCES [dbo].[Клиент] ([Номер клиента])

GO

ALTER TABLE [dbo].[Dogovor] CHECK CONSTRAINT [FK\_Договор\_Клиент]

GO

ALTER TABLE [dbo].[Dogovor] WITH CHECK ADD CONSTRAINT [FK\_Договор\_Тариф] FOREIGN KEY([Код тарифа FK])

REFERENCES [dbo].[Тариф] ([Код тарифа])

GO

ALTER TABLE [dbo].[Dogovor] CHECK CONSTRAINT [FK\_Договор\_Тариф]

GO

1. **Звонок**

USE [operator]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Звонок] Script Date: 17.11.2022 8:30:13 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Звонок](

[Дата и время] [date] NULL,

[Длительность] [int] NULL,

[Стоимость] [money] NULL,

[Номер договора FK] [int] NULL,

[Код типа звонка FK] [int] NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Звонок] WITH CHECK ADD CONSTRAINT [FK\_Звонок\_Тип звонка] FOREIGN KEY([Номер договора FK])

REFERENCES [dbo].[Dogovor] ([Номер договора])

GO

ALTER TABLE [dbo].[Звонок] CHECK CONSTRAINT [FK\_Звонок\_Тип звонка]

GO

ALTER TABLE [dbo].[Звонок] WITH CHECK ADD CONSTRAINT [FK\_Звонок\_Тип звонка1] FOREIGN KEY([Код типа звонка FK])

REFERENCES [dbo].[Тип звонка] ([Код типа звонка])

GO

ALTER TABLE [dbo].[Звонок] CHECK CONSTRAINT [FK\_Звонок\_Тип звонка1]

GO

1. **Клиент**

USE [operator]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Клиент] Script Date: 17.11.2022 8:31:22 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Клиент](

[Номер клиента] [int] NOT NULL,

[ФИО] [varchar](50) NULL,

[Баланс] [money] NULL,

CONSTRAINT [PK\_Клиент] PRIMARY KEY CLUSTERED

(

[Номер клиента] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

1. **Тариф**

USE [operator]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Тариф] Script Date: 17.11.2022 8:32:09 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Тариф](

[Минута межгород стоимость] [money] NULL,

[Минута международная стоимость] [money] NOT NULL,

[Название тарифа] [varchar](50) NOT NULL,

[Стоимость перехода] [money] NOT NULL,

[Код типа тарифа FK] [int] NOT NULL,

[Код тарифа] [int] NOT NULL,

[Год\_начала] [int] NULL,

CONSTRAINT [PK\_Тариф\_1] PRIMARY KEY CLUSTERED

(

[Код тарифа] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Тариф] WITH CHECK ADD CONSTRAINT [FK\_Тариф\_Тип тарифа] FOREIGN KEY([Код типа тарифа FK])

REFERENCES [dbo].[Тип тарифа] ([Код типа тарифа])

GO

ALTER TABLE [dbo].[Тариф] CHECK CONSTRAINT [FK\_Тариф\_Тип тарифа]

GO

1. **Тип звонка**

USE [operator]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Тип звонка] Script Date: 17.11.2022 8:32:53 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Тип звонка](

[Название типа звонка] [varchar](50) NULL,

[Код типа звонка] [int] NOT NULL,

CONSTRAINT [PK\_Тип звонка] PRIMARY KEY CLUSTERED

(

[Код типа звонка] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

1. **Тип тарифа**

USE [operator]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Тип тарифа] Script Date: 17.11.2022 8:33:19 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Тип тарифа](

[Название типа тарифа] [varchar](50) NULL,

[Код типа тарифа] [int] NOT NULL,

CONSTRAINT [PK\_Тип тарифа] PRIMARY KEY CLUSTERED

(

[Код типа тарифа] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

## Решения по программному обеспечению

Решения эскизного проектирования по программному обеспечению представлены проектом классов (диаграмма классов и описание полей и методов классов), моделью инициализации объектов (диаграмма инициализации) и моделью взаимодействия (диаграммы последовательности).

### Классы

В результате детального проектирования, получены следующие классы:

### BLL. Interfaces.IDbCrud

List<КлиентModel> GetAllКлиент() – получение списка всех клиентов.

КлиентModel GetКлиент(int clvId) – получение клиента по Id.

void CreateКлиент(КлиентModel cl) – создание клиента.

void UpdateКлиент(КлиентModel cl) – обновление клиента.

void DeleteКлиент(int id) – удаление клиента по Id.

List<ТарифModel> GetAllТариф() – получение списка всех тарифов.

ТарифModel GetТариф(int trfId) – получение тарифа по Id.

void CreateТариф(ТарифModel trf) – создание тарифа.

void UpdateТариф(ТарифModel trf) – обновление тарифа.

void DeleteТариф(int id) – удаление тарифа по Id.

List<DogovorModel> GetAllDogovor() – получение списка всех договоров.

DogovorModel GetDogovor(int dgvId) – получение договора по Id.

void CreateDogovor(DogovorModel dgv) – создание договора.

void UpdateDogovor(DogovorModel dgv) – обновление договора.

void DeleteDogovor(int id) – удаление договора по Id.

### BLL. Interfaces.IReportService

List<Dogovors> procedd8(DateTime start, DateTime end) – процедура выполнения отчета поиска договоров по временным промежуткам.

List<BLL.Models.CallModel> procedd81(int kod) – процедура выполнения отчета поиска совершенным договором звонков.

List<BLL.Models.КлиентModel> procedd812(string s) – процедура выполнения отчета поиска ФИО клиента по букве.

### BLL.Models.Call

class CallModel – класс совершенных звонков.

public DateTime Дата\_и\_время – дата и время совершенного звонка.

public int Длительность – длительность совершенного звонка.

public decimal Стоимость – полная стоимость совершенного звонка

public int Номер\_договора\_FK – номер договора, по которому был совершен звонок.

public int Код\_типа\_звонка\_FK – код типа, к которому относится звонок.

### BLL.Models.Dogovor

### public DogovorModel - класс договоров.

### public DateTime Дата\_заключения – дата заключения договора.

public int Номер\_договора – номер договора.

public string Номер\_телефона – номер телефона, который прикреплен к договору.

public string Серийный\_номер\_сим\_карты – серийный номер сим карты, которая прикреплена к договору.

public DateTime Дата\_расторжения – дата расторжения договора.

public int Код\_тарифа\_FK – код тарифа, который прикреплен к договору.

public int Номер\_клиента\_FK – номер клиента договора.

### BLL.Models.Client

class КлиентModel – класс клиентов.

public int Номер\_клиента – номер клиента.

public string ФИО – фамилия, имя, отчество клиента.

public decimal? Баланс – баланс клиента.

### BLL.Models.Тариф

class ТарифModel – класс тарифов.

public decimal? Минута\_межгород\_стоимость – стоимость минуты звонка, совершенного между городами.

public decimal Минута\_международная\_стоимость – стоимость минуты звонка, совершенного по миру.

public string Название\_тарифа – название тарифа.

public decimal Стоимость\_перехода – название стоимости.

public int Код\_типа\_тарифа\_FK – код типа, к которому относится тариф.

public int Код\_тарифа – код тарифа.

public int? Год\_начала – год открытия тарифа.

### BLL.Services.ReportServices

Сервисы процедур:

List<Dogovors> procedd8(DateTime start, DateTime end) – процедура выполнения отчета поиска договоров по временным промежуткам.

List<BLL.Models.CallModel> procedd81(int kod) – процедура выполнения отчета поиска совершенным договором звонков.

List<BLL.Models.КлиентModel> procedd812(string s) – процедура выполнения отчета поиска ФИО клиента по букве.

### BLL.DBDataOperation

List<КлиентModel> GetAllКлиент() – получение списка всех клиентов.

КлиентModel GetКлиент(int clvId) – получение клиента по Id.

void CreateКлиент(КлиентModel cl) – создание клиента.

void UpdateКлиент(КлиентModel cl) – обновление клиента.

void DeleteКлиент(int id) – удаление клиента по Id.

List<ТарифModel> GetAllТариф() – получение списка всех тарифов.

ТарифModel GetТариф(int trfId) – получение тарифа по Id.

void CreateТариф(ТарифModel trf) – создание тарифа.

void UpdateТариф(ТарифModel trf) – обновление тарифа.

void DeleteТариф(int id) – удаление тарифа по Id.

List<DogovorModel> GetAllDogovor() – получение списка всех договоров.

DogovorModel GetDogovor(int dgvId) – получение договора по Id.

void CreateDogovor(DogovorModel dgv) – создание договора.

void UpdateDogovor(DogovorModel dgv) – обновление договора.

void DeleteDogovor(int id) – удаление договора по Id.

### DAL.Interfaces.IdbRepos

IRepository<Клиент> Клиент – репозиторий клиента.

IRepository<Тариф> Тариф – репозиторий тарифа.

IRepository<Dogovor> Dogovor – репозиторий договора.

IReportsRepository Reports – репозиторий отчета

### DAL.Interfaces.IReportsRepository

List<Dogovors> procedd8(DateTime start, DateTime end) – репозиторий процедуры выполнения отчета поиска договоров по временным промежуткам.

List<BLL.Models.CallModel> procedd81(int kod) – репозиторий процедуры выполнения отчета поиска совершенным договором звонков.

List<BLL.Models.КлиентModel> procedd812(string s) – репозиторий процедуры выполнения отчета поиска ФИО клиента по букве.

### DAL.Interfaces.IRepository

List<T> GetList() - получение всех объектов.

T GetItem(int id) - получение одного объекта по id.

void Create(T item) - создание объекта.

void Update(T item) - обновление объекта.

void Delete(int id) - удаление объекта по id.

### DAL.Models.Call

Загружено через CodeFirst.

class CallModel – класс совершенных звонков.

public DateTime Дата\_и\_время – дата и время совершенного звонка.

public int Длительность – длительность совершенного звонка.

public decimal Стоимость – полная стоимость совершенного звонка

public int Номер\_договора\_FK – номер договора, по которому был совершен звонок.

public int Код\_типа\_звонка\_FK – код типа, к которому относится звонок.

### DAL.Models.Dogovor

Загружено через CodeFirst.

public DogovorModel - класс договоров.

public DateTime Дата\_заключения – дата заключения договора.

public int Номер\_договора – номер договора.

public string Номер\_телефона – номер телефона, который прикреплен к договору.

public string Серийный\_номер\_сим\_карты – серийный номер сим карты, которая прикреплена к договору.

public DateTime Дата\_расторжения – дата расторжения договора.

public int Код\_тарифа\_FK – код тарифа, который прикреплен к договору.

public int Номер\_клиента\_FK – номер клиента договора.

### DAL.Models.Client

Загружено через CodeFirst.

class КлиентModel – класс клиентов.

public int Номер\_клиента – номер клиента.

public string ФИО – фамилия, имя, отчество клиента.

public decimal? Баланс – баланс клиента.

### DAL.Models.Тариф

Загружено через CodeFirst.

class ТарифModel – класс тарифов.

public decimal? Минута\_межгород\_стоимость – стоимость минуты звонка, совершенного между городами.

public decimal Минута\_международная\_стоимость – стоимость минуты звонка, совершенного по миру.

public string Название\_тарифа – название тарифа.

public decimal Стоимость\_перехода – название стоимости.

public int Код\_типа\_тарифа\_FK – код типа, к которому относится тариф.

public int Код\_тарифа – код тарифа.

public int? Год\_начала – год открытия тарифа.

### DAL.Models.Тип\_звонка

class Тип\_звонка – класса типа звонка: международный или междугородний.

public string Название\_типа\_звонка – название типа звонка.

public int Код\_типа\_звонка – код типа совершенного звонка.

### DAL.Models.Тип\_тарифа

class Тип\_тарифа – класс типа тарифа.

public string Название\_типа\_тарифа – название типа тарифа.

public int Код\_типа\_тарифа – код типа тарифа.

### DAL.Repository.DbReposSQL

Определяет все репозитории:

IRepository<Клиент>

IReportsRepository

IRepository<Dogovor>

IRepository<Тариф>

### DAL.Repository.DogovorRepository

Определяет основные методы работы с договором:

GetList() – получение списка.

GetItem(int id) – получение по Id.

Create(Dogovor dogovor) – создание нового договора.

Update(Dogovor dogovor) – обновление договора.

Delete(int id) – удаление договора по Id.

### DAL.Repository.ReportRepository

Содержит описание содержимого процедур:

List<Dogovors> procedd8(DateTime start, DateTime end) – процедура выполнения отчета поиска договоров по временным промежуткам.

List<BLL.Models.CallModel> procedd81(int kod) – процедура выполнения отчета поиска совершенным договором звонков.

List<BLL.Models.КлиентModel> procedd812(string s) – процедура выполнения отчета поиска ФИО клиента по букве.

### DAL.Repository.КлиентRepository

Определяет основные методы работы с клиентом:

GetList() – получение списка.

GetItem(int id) – получение по Id.

Create(Dogovor dogovor) – создание нового клиента.

Update(Dogovor dogovor) – обновление клиента.

Delete(int id) – удаление клиениа по Id.

### DAL.Repository.ТарифRepository

Определяет основные методы работы с тарифом:

GetList() – получение списка.

GetItem(int id) – получение по Id.

Create(Dogovor dogovor) – создание нового тарифа.

Update(Dogovor dogovor) – обновление тарифа.

Delete(int id) – удаление тарифа по Id.

### Wpf.Commands.RelayCommand

public class RelayCommand – класс определения команд.

### Wpf.ModelView.MainFormViewModel

Методы для основного окна:

public DogovorModel SelectedDogovor – выделение строки договора.

public КлиентModel SelectedКлиент – выделение строки клиента.

public ТарифModel SelectedТариф – выделение строки тарифа.

public RelayCommand RemoveClientCommand – удаление клиента.

public RelayCommand RemoveTarifCommand – удаление тарифа.

public RelayCommand RemoveDogovorCommand – удаление договора.

public RelayCommand UpdateDogovorCommand – обновление данных договора.

public RelayCommand UpdateClientCommand – обновление данных клиента.

public RelayCommand UpdateTarifCommand – обновление данных тарифа

public void loadTarif(System.Windows.Controls.DataGrid) – вывод всех тарифов.

public void loadClient(System.Windows.Controls.DataGrid) – вывод всех клиентов.

public void loadDogovor(System.Windows.Controls.DataGrid) – вывод всех договоров.

public RelayCommand ReportCommand – вывод отчета по договорам.

public RelayCommand Report2Command – вывод отчета по звонкам.

public RelayCommand Report3Command – вывод отчета по клиентам.

public RelayCommand Print – печать.

### Wpf.ModelView.Window1ViewModel

Методы для окна добавления договора:

public RelayCommand AddDogovorCommand – команда добавления для кнопки «Сохранить»

public void LoadComboboxTarif – вывод номеров всех тарифов в комбобокс.

public void LoadComboboxClient – вывод номеров всех клиентов в комбобокс.

### Wpf.ModelView.Window2ViewModel

Методы для окна добавления клиента:

public RelayCommand AddClientCommand – команда добавления для кнопки «Сохранить».

### Wpf.ModelView.Window3ViewModel

Методы для окна добавления тарифа:

public RelayCommand AddTarifCommand – команда добавления для кнопки «Сохранить».

### Wpf.ModelView.Window6ViewModel

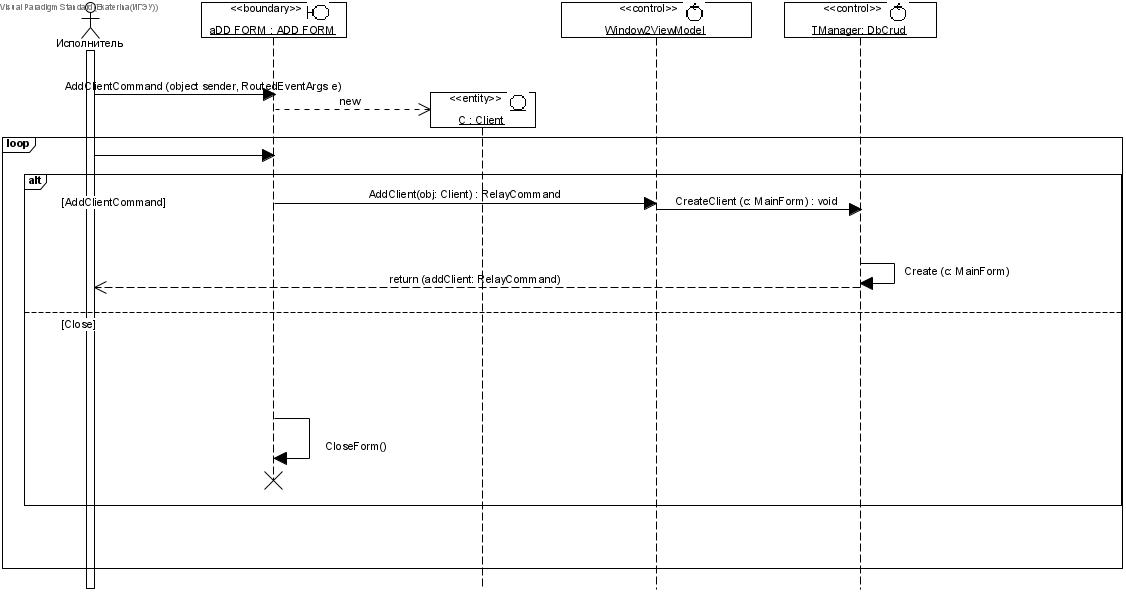
Методы для окна составления отчета совершенных звонков по номеру договора:

public void LoadComboboxDogovor – вывод номеров всех договоров в комбобокс.

### Wpf.ModelView.Util.NinjectRegistrations

class NinjectRegistration – создает подключение к ядру.

### Диаграмма последовательности для варианта «Добавить клиента»

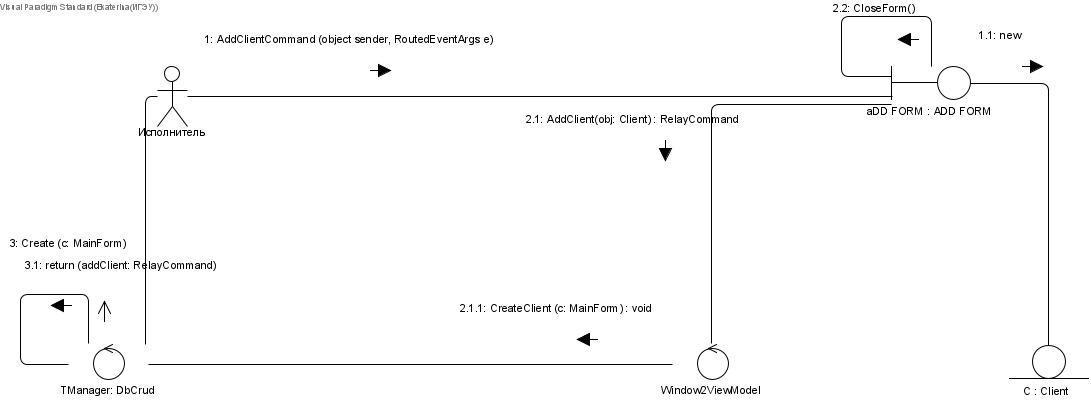


**Рис. 3.9.4 Диаграмма последовательности**

Пользователь нажимает кнопку AddClientCommand и открывается форма – AddForm, создается новая сущность Клиента. В альтернативных потоках:

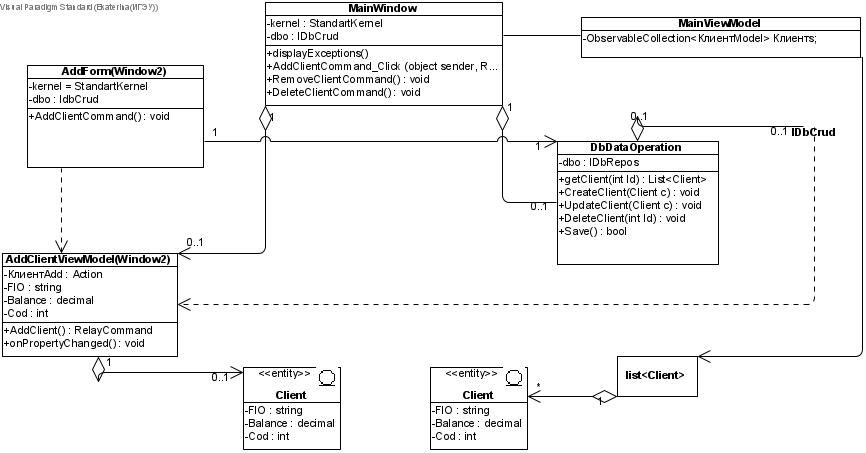
1. Добавление клиента, после нажатия кнопки «Сохранить» на форме, поступает команда AddClient: relayCommand, от контроллера – ViewModel идет обращение к IDbCrud, который создает клиента методом CreateClient и возвращает значение обратно Пользователю.
2. Закрытие формы, после нажатия кнопки «Закрыть» форма закрывается и ее жизненный цикл заканчивается.

### Диаграмма коммуникации для варианта «Добавить клиента»



**Рис. 3.9.5 Диаграмма коммуникации**

### Диаграмма классов для варианта «Добавить клиента»

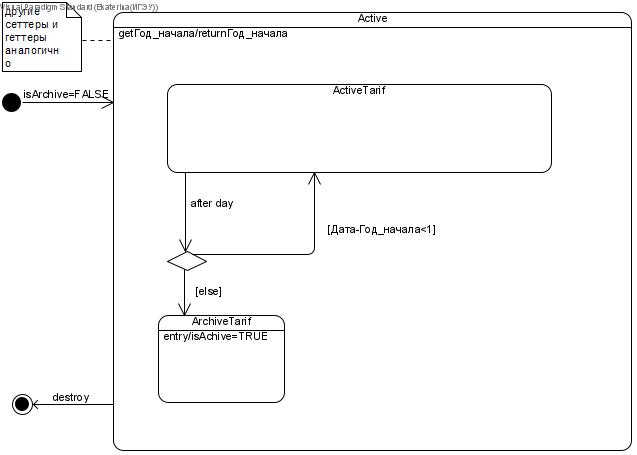


**Рис. 3.9.6 Диаграмма классов**

После открытия AddForm на MainWindow, она обращается к контроллеру – AddClientViewModel, который создает новую сущность.

В то же время в MainViewModel хранится список всех клиентов.

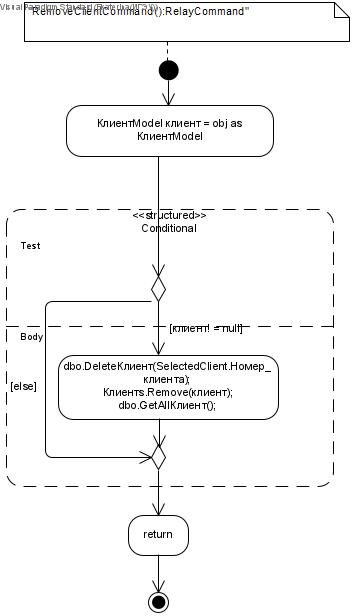
### Диаграмма состояний для метода «Архивация тарифа»



**Рис. 3.9.7 Диаграмма состояний**

У Тарифа есть 2 состояния – активный и архивированный. После того, как проходит один день, проверяется условие: если от года, когда тариф был активирован, прошел целый год, то он архивируется, иначе остается активным.

### Диаграмма деятельности для варианта «Удалить клиента»



**Рис. 3.9.8 Диаграмма деятельности**

Удаление клиента происходит при условии, если он не равен 0.

# Мероприятия по подготовке к вводу системы в действие

Для ввода Системы в действие необходимо:

1. подготовить у Заказчика всё необходимое техническое обеспечение;
2. установить на сервер и клиентские ПК системное, базовое и прикладное ПО;
3. установить на сервер БД;
4. провести испытания системы.

## Мероприятия по приведению информации к виду, пригодному для обработки на ЭВМ

Для ввода системы в эксплуатацию необходимо выполнить первичное наполнение БД:

1. ввести в БД сведения о Сотрудниках,
2. ввести в БД сведения о деловых партнёрах,
3. Подготовить биллинговую систему.

## Мероприятия по обучению и проверке квалификации персонала

Необходимо научить Операторов работе с клиентами, тарифами и договорами, как их добавлять, изменять, удалять и как запрашивать по ним статистику.

## Мероприятия по созданию необходимых подразделений и рабочих мест

Новые подразделения не создаются.

## Мероприятия по изменению объекта автоматизации

Никаких изменений не требуется.

# Источники разработки

При разработке эскизного проекта использовались следующие источники:

1. ГОСТ 34.003-90. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения.
2. ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
3. ГОСТ 34.601-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
4. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
5. РД 50-34.698.90. Методические указания. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов.
6. РД Гостехкомиссии России. Безопасность информационных технологий. Критерии оценки безопасности информационных технологий.- 2002 г.
7. Практическая программная инженерия на основе учебного примера / Л.А. Мацяшек, Б.Л. Лионг. – М.: БИНОМ. Лаборатория знаний, 2009. – 956 с.
8. Documents associated with. Unified Modeling Language (UML), v2.4.1. Release date: August 2011.- http://www.omg.org/spec/UML/2.4.1.