Содержание

[Введение 4](#_Toc151737033)

[1 Цель и задачи УЧЕБНОЙ практики. 5](#_Toc151737034)

[1.1 Основные требования по технике безопасности во время прохождения учебной практики 6](#_Toc151737035)

[1.2 Нормативно-правовые документы 7](#_Toc151737036)

[2 Предметная область. Анализ предметной области. Постановка задачи. 9](#_Toc151737037)

[2.1 Описание предметной области 9](#_Toc151737038)

[2.2 Группы пользователей информационной системы 9](#_Toc151737039)

[2.3 Основные требования, предъявляемые к информационной системе 10](#_Toc151737040)

[3 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «Охрана квартир» 11](#_Toc151737041)

[3.1 Моделирование некоторых динамических аспектов системы 11](#_Toc151737042)

[3.1.1 Создание диаграммы вариантов использования 12](#_Toc151737043)

[3.1.2 Создание диаграммы деятельности 14](#_Toc151737044)

[3.2 Разработка функциональной структуры информационной системы 15](#_Toc151737045)

[3.3 Разработка макета информационной системы 15](#_Toc151737046)

[4 РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ «Охрана квартир» 16](#_Toc151737047)

[4.1 Моделирование и разработка базы данных 16](#_Toc151737048)

[4.2. Разработка программных модулей информационной системы «Охрана квартир» 21](#_Toc151737049)

[5 РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ 22](#_Toc151737050)

[5.1 Назначение эксперимента. Выбор и обоснование методики проведения тестирования 22](#_Toc151737051)

[5.2 Технология тестирования 23](#_Toc151737052)

[5.3 Результаты проведения тестирования 23](#_Toc151737053)

[ЗАКЛЮЧЕНИЕ 24](#_Toc151737054)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 25](#_Toc151737055)

[ПРИЛОЖЕНИЕ A 26](#_Toc151737056)

[ПРИЛОЖЕНИЕ Б 30](#_Toc151737057)

# Введение

Каждому человеку нужна охрана своего имущества и своей квартиры. Современные вызовы в области безопасности жилья подчеркивает необходимость эффективных методов и технологий для защиты жилищных помещений. Один из таких подходов — разработка прикладного программного обеспечения для отделов вневедомственной охраны квартир. Это программное обеспечение будет предоставлять инновационные решения, направленные на повышение эффективности мер по обеспечению безопасности от злоумышленников. Разработка прикладного программного обеспечения для отдела вневедомственной охраны квартир представляет собой важный шаг в направлении повышения общей безопасности общества.

Для освоения основного вида профессиональной деятельности: «Разработка программного обеспечения для компьютерных систем» и соответствующих профессиональных компетенций:

ПК 1.1 - Выполнять разработку спецификаций отдельных компонент

ПК 1.2 - Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля

ПК 1.3 - Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 - Выполнять тестирование программных модулей

ПК 1.5 - Осуществлять оптимизацию программного модуля

ПК 1.6 - Разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций

# Цель и задачи УЧЕБНОЙ практики.

Целью учебной практики является овладение видом профессиональной деятельности Разработка модулей программного обеспечения для компьютерных систем, общими и профессиональными компетенциями по специальности.

Основные задачи учебной практики:

* систематизация, обобщение, закрепление и углубление знаний и умений;
* формирование у студентов знаний, умений и навыков, профессиональных компетенций, профессионально значимых личностных качеств;
* развитие профессионального интереса, формирование мотивационно-целостного отношения к профессиональной деятельности, готовности к выполнению профессиональных задач в соответствии с нормами морали, профессиональной этики и служебного этикета.

Учебная практика профессионального модуля ПМ 01 Разработка модулей программного обеспечения для компьютерных систем проводится в кабинетах информационно - вычислительного центра техникума преподавателями общепрофессиональных дисциплин профессионального цикла.

В период выполнения заданий учебной практики профессиональных модулей на студентов распространяются требования охраны труда и правила внутреннего распорядка, действующие в ГБПОУ ИО «АПЭТ»

Электронная охрана квартир позволяет добиться безопасности от злоумышленников в районе. Программный продукт представляет собой универсальное программное обеспечение, предназначенное для деятельности отдела вневедомственной охраны квартир, для удобной эксплуатации в целях предоставления информации.

Программный продукт предназначен для индивидуального применения, то есть предполагает использование одним пользователем в целях получения информации. В программном продукте можно будет заключать договор на охрану квартир.

## Основные требования по технике безопасности во время прохождения учебной практики

К работе в компьютерном классе допускаются лица, ознакомленные с данной инструкцией по технике безопасности и правилам поведения. Необходимо неукоснительно соблюдать правила по технике безопасности, т.к. нарушение этих правил может привести к поражению электрическим током, вызвать возгорание и навредить вашему здоровью.

* Работа обучающихся в компьютерном классе разрешается только в присутствии преподавателя (лаборанта).
* Во время занятий посторонние лица могут находиться в классе только с разрешения преподавателя.
* Во время перемен между уроками проводится обязательное проветривание компьютерного кабинета с обязательным выходом обучающихся из класса.

Каждый обучающийся в ответе за состояние своего рабочего места и сохранность размещенного на нем оборудования.

Обучающийся, допустивший нарушение по технике безопасности и правилам поведения в компьютерном классе, может быть привлечен к дисциплинарной ответственности. Если нарушение техники безопасности связано с причинением имущественного ущерба, обучающийся несет и материальную ответственность в установленном законом порядке.

## 1.2 Нормативно-правовые документы

Оформление и содержание технического задания должно соответствовать требованиям стандарта «ГОСТ 19.201-78. ЕСПД. Общие положения.

Техническое задание оформляют в соответствии с ГОСТ 19.106-78 на листах формата 11 и 12 по ГОСТ 2.301-68, как правило, без заполнения полей листа. Номера листов (страниц) проставляются в верхней части листа над текстом.

Лист утверждения и титульный лист оформляют в соответствии с ГОСТ 19.104-78.

Информационную часть (аннотацию и содержание), лист регистрации изменений допускается в документ не включать.

Для внесения изменений или дополнений в техническое задание на последующих стадиях разработки про1раммы или программного изделия выпускают дополнение к нему.

Согласование и утверждение дополнения к техническому заданию проводят в том же порядке, который установлен для технического задания.

Техническое задание должно содержать следующие разделы:

* введение;
* основания для разработки;
* назначение разработки;
* требования к программе или программному изделию;
* требования к программной документации;
* технико-экономические показатели;
* стадии и этапы разработки;
* порядок контроля и приемки;
* в техническое задание допускается включать приложения.

В зависимости от особенностей программы или программного изделия допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них.

# Предметная область. Анализ предметной области. Постановка задачи.

## 2.1 Описание предметной области

В рамках учебной практики было выдано следующее задание: разработать прикладное программное обеспечение деятельности отдела вневедомственной охраны квартир. Этот отдел обеспечивает электронную охрану квартир граждан в одном районе города. Для установки охранной сигнализации требуется наличие квартирного телефона. Один гражданин может заключить договор на охрану нескольких квартир. Из-за ложных срабатываний сигнализации возможно несколько выездов патрульных экипажей по одной квартире. На владельца квартиры, вовремя не отключившего сигнализацию после своего прихода домой, налагается штраф, величина которого оговаривается при заключении договора охраны. Если отдел вневедомственной охраны не уберег имущество владельца квартиры, то он выплачивает пострадавшему заранее оговоренную сумму. От величины этой суммы зависит размер ежемесячной оплаты за охрану квартиры.

Необходимо создать базу данных, обеспечивающая деятельность отдела вневедомственной охраны квартир.

## 2.2 Группы пользователей информационной системы

В данной информационная системе можно выделить несколько пользователей:

* Случайный пользователь (Гость) – данный пользователь не может взаимодействовать с информационной системой.
* Конечный пользователь (потребитель информационной системы) – лицо или группа лиц, в интересах которых работает ИС. Данный пользователь был зарегистрирован в информационной системе и, как правило, не является программистом
* Коллектив специалистов (персонал информационной системы) – включающий администратора, командир экипажа и оператора.

## 2.3 Основные требования, предъявляемые к информационной системе

Данная информационная система должна соответствовать следующим требованиям:

* Гибкость – данная информационная система должна иметь возможность добавления новых таблиц.
* Надежность – данная информационная система должна выполнять свои функции.
* Эффективность – данная информационная система должна максимально быстро выполнять свои функции.
* Безопасность – данная информационная система должна быть защищен от взломов и иметь возможность восстановления и создания резерва.

В итоге данная информационная система должна

# ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «Охрана квартир»

Проектированием информационных систем называется многоступенчатый процесс их создания и/или модернизации путём применения упорядоченной совокупности методологий и инструментария. Проектирование (в отличие от моделирования) предполагает работу с пока несуществующим объектом и направлено на создание информационной системы в области:

* обработки объектов будущей базы данных,
* написания программ (в том числе – отчётных и экранных форм), обеспечивающих выполнение запросов к данным,
* выполнения учёта функционирования конкретной среды (технологии).

Если выделять стадию проектирования информационных систем в качестве отдельного этапа, то его можно разместить между этапами анализа и разработки. Однако на практике чёткое разделение на этапы, как правило, затруднено или невозможно, поскольку проектирование, формально начинаясь с определения цели проекта, часто продолжается на стадиях тестирования и реализации.

## 3.1 Моделирование некоторых динамических аспектов системы

Диаграммы деятельности применяются для моделирования динамических аспектов поведения системы. Эти динамические аспекты могут включать деятельность на любом уровне абстракции в любом представлении системной архитектуры, включая классы (в том числе активные), интерфейсы, компоненты и узлы.

Использовать диаграммы деятельности для моделирования некоторых динамических аспектов системы вы можете в контексте почти любого моделируемого элемента. Однако чаще вы будете прибегать к таким диаграммам в контексте всей системы, подсистемы, операции или класса. Диаграмму деятельности можно присоединить к варианту использования, чтобы моделировать сценарий, и к кооперации, чтобы моделировать динамические аспекты поведения совокупности объектов.

При разработке программного обеспечения заблаговременное планирование и моделирование значительно упрощают программирование. На этапе создания концептуальной модели для описания функционального назначения используется диаграмма вариантов использования, а для описания логики процедур и бизнес-процессов используется диаграмма деятельности.

### 3.1.1 Создание диаграммы вариантов использования

Перед началом разработки информационной системы необходимо создать визуальное представление о вариантах использования разрабатываемой системы.

Диаграмма вариантов использования является концептуальным представлением системы в процессе ее проектирования и разработки. Диаграмма вариантов использования описывает функциональное назначение системы или, другими словами, то, что система будет делать в процессе своего функционирования.

Цели построения диаграммы вариантов использования:

* определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования;
* сформулировать общие требования к функциональному проектированию системы;
* разработать исходную концептуальную модель системы для ее последующей реализации;
* подготовить документацию для взаимодействия разработчика системы с ее заказчиком и пользователями.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью, так называемых, вариантов использования. При этом актером или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик.

В свою очередь, вариант использования служит для описания сервисов, которые система предоставляет актеру. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером.

В Приложении «А» на рисунке 4 представлена диаграмма вариантов использования для информационной системы «Охрана квартир». На этой диаграмме представлены следующие актеры:

1. Случайный пользователь (Гость) – может добавлять записи в таблицы: Client.
2. Конечный пользователь – может просматривать и читать таблицы: Capture, Contract, Flat, House, Prolonging, Calling. Также может добавлять записи в таблицы: Prolonging, Flat, House, Contract.
3. Командир экипажа – может просматривать и читать таблицы: House, Contract, Flat, Client, Flat-House, Settlement. И добавлять, изменять и удалять записи в таблице Capture, Calling.
4. Оператор – может изменять, удалять и добавлять данные во всех таблицах.
5. Администратор – может добавлять, удалять и изменять все таблицы в информационной системе. Также он может добавлять новых пользователей и разграничивать права доступа к базе данных. Это необходимо для того, чтобы сотрудники, не имеющие определенных полномочий, не могли получить доступ к различным объектам и данным.

### 3.1.2 Создание диаграммы деятельности

При моделировании поведения системы возникает необходимость не только представить процесс изменения ее состояний, но и детализировать особенности алгоритмической и логической реализации выполняемых системой операций. Для этого применяется диаграмма деятельности.

Диаграмма деятельности — диаграмма, описывающая динамические аспекты системы. Диаграмма деятельности UML позволяет более детально визуализировать конкретный случай использования. Это поведенческая диаграмма, которая иллюстрирует поток деятельности через систему.

В Приложении «Б» на рисунке 5 находится диаграмма деятельности «Гость», которая показывает последовательность действий гостя, необходимых для достижения той или иной цели.

В Приложении «Б» на рисунке 5 находится диаграмма деятельности «Гость», которая показывает последовательность действий гостя, необходимых для достижения той или иной цели.

В Приложении «Б» на рисунке 5 находится диаграмма деятельности «Гость», которая показывает последовательность действий гостя, необходимых для достижения той или иной цели.

В Приложении «Б» на рисунке 5 находится диаграмма деятельности «Гость», которая показывает последовательность действий гостя, необходимых для достижения той или иной цели.

В Приложении «Б» на рисунке 5 находится диаграмма деятельности «Гость», которая показывает последовательность действий гостя, необходимых для достижения той или иной цели.

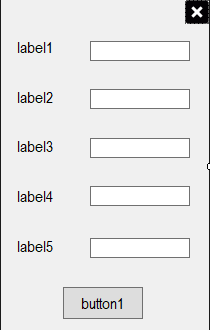
## 3.2 Разработка функциональной структуры информационной системы

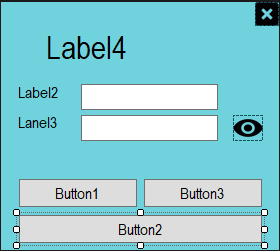
Принципы модульного программирования во многом схожи с нисходящим проектированием, сначала определяется состав и подчиненность функции, а затем набор программных модулей, реализующих эти функции. Функции верхнего уровня обеспечиваются главным модулем, он управляет выполнением нижестоящих функций, которым соответствуют подчиненные модули. При определении набора модулей необходимо учитывать:

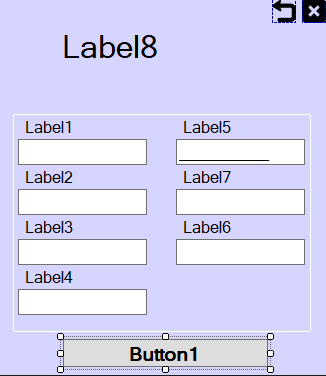
* каждый модуль вызывается на выполнение вышестоящим;
* принятие основных решений в алгоритме выносится на максимально высокий уровень по иерархии уровней;
* для использования одной и той же функции создается один модуль, который вызывается на выполнение по мере необходимости.

Карта навигации по информационной системе представлена в приложении «Б» на рисунке 3 представлена карта переходов по информационной системе «Охрана квартир».

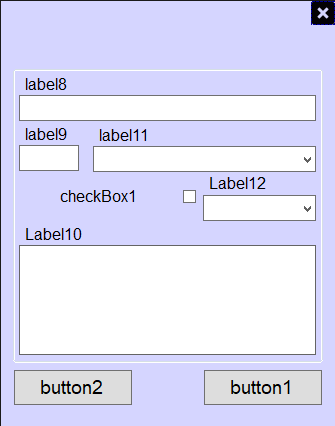
## 3.3 Разработка макета информационной системы

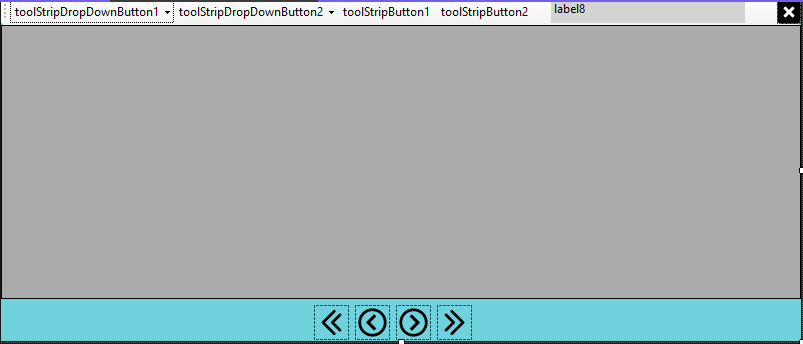












# РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ «Охрана квартир»

Прототипирование — это один из этапов разработки, который заключается в продумывании содержания и расположения важных элементов интерфейса. Прототип — это моделирование конечного продукта. Это интерактивный макет, который может иметь любую степень точности. Основная цель создания прототипов – проверить, насколько последователен путь пользователя, и выявить препятствия, которые могут возникнуть в процессе его взаимодействия с продуктом.

Прототипы не только позволяют проверить удобство разрабатываемого продукта до начала написания кода, они также приводят к неожиданным открытиям и новым идеям, которые могут вывести проектируемый продукт на новый уровень.

## 4.1 Моделирование и разработка базы данных

Исходя из анализа предметной области и набора, данных предоставленные вариантом учебной практики из Приложения «А» (рис. 1), можно выделить 11 типа сущностей (таблиц): Client, Flat, House, Contract, Tax, Compensation, Prolonging, Capture, Settlement, Flat-House, Calling.

**Описание сущностей**:

Один клиент, может иметь несколько квартир, поэтому между объектами Client и Settlement имеется связь «один-ко-многим». В одном доме может быть несколько квартир, поэтому между объектами Flat-House и House имеется связь «один-ко-многим». На одну квартиру можно заключить несколько договоров, поэтому между объектами Settlement лицо и Contract имеется связь «один-ко-многим». На одну квартиру может выехать несколько бригад на захват, поэтому между объектами Capture и Contract имеется связь «один-ко-многим». Договор можно несколько раз продлить, поэтому между объектами Contract и Prolonging имеется связь «один-ко-многим».

* к объекту Client относятся такие характеристики как: SecondName, FirstName, ThirdName, Phone, Address и Registr, который является полем первичного ключа;
* к объекту Flat относится: AddressFlat, Floor, TypeDoor, TypeBalcony, Balcony, Plan и FlatID, который является полем первичного ключа;
* к объекту House относится: Floors, Key, TypeHouse и HouseID, который является первичным ключом;
* к объекту Settlement относится: Registr, который является полем внешнего ключа и соединен с полем Registr в таблице Client, FlatID, который является полем внешнего ключа и соединен с полем FlatID в таблице Flat, SettlementID, который является полем первичного ключа;
* к объекту Flat-House относится: FlatID, который является полем внешнего ключа и соединен с полем FlatID в таблице Flat, HouseID, который является полем внешнего ключа и соединен с полем HouseID в таблице House, Flat-HouseID, который является полем первичного ключа;
* к объекту Contract относится: ContractName, DataStart, StopDate, Cost, SettlementID, который является полем внешнего ключа и соединен с полем SettlementID в таблице Settlement, TreatyID, который является полем первичного ключа;
* к объекту Prolonging относится: Prolong,Comment, TreatyID, который является полем внешнего ключа и соединен с полем TreatyID в таблице Contract, ProlongingID, который является полем первичного ключа;
* к объекту Capture относится: PatrollID, Chief, Brand, Document, ActionID, который является полем первичного ключа;
* к объекту Calling относится: DateTime, False, TreatyID, который является полем внешнего ключа и соединен с полем TreatyID в таблице Contract, ActionID, который является полем внешнего ключа и соединен с полем ActionID в таблице Capture, CallingID, который является полем первичного ключа;

В приложении «А» на рисунке 2 представлена логическая модель данных, созданная по этому набору данных.

С помощью раздела «Создание» и конструктора таблиц MS Visual Studio созданы все таблицы и поля, представленные на логической модели данных.

Названия полей и их типы данных представлены в таблицах 2-11.

Таблица 1 – Поля таблицы «Client»

| Название поля | Тип данных |
| --- | --- |
| Registr | Числовой |
| SecondName | Текстовый |
| FirstName | Текстовый |
| ThirdName | Текстовый |
| Address | Текстовый |
| Phone | Текстовый |

Таблица 2 – Поля таблицы «Flat»

| Название поля | Тип данных |
| --- | --- |
| FlatID | Числовой |
| AddressFlat | Текстовый |
| Floor | Текстовый |
| TypeDoor | Текстовый |
| Balcony | Логический |
| TypeBalcony | Текстовый |
| Plane | Image |

Таблица 3 – Поля таблицы «House»

| Название поля | Тип данных |
| --- | --- |
| HouseID | Числовой |
| Floors | Текстовый |
| Key | Логический |
| TypeHouse | Текстовый |

Таблица 4 – Поля таблицы «Flat-House»

| Название поля | Тип данных |
| --- | --- |
| Flat-HouseID | Числовой |
| HouseID | Числовой |
| FlatID | Числовой |

Таблица 5 – Поля таблицы «Settlement»

| Название поля | Тип данных |
| --- | --- |
| SettlementID | Числовой |
| Registr | Числовой |
| FlatID | Числовой |

Таблица 6 – Поля таблицы «Contract»

| Название поля | Тип данных |
| --- | --- |
| TreatyID | Числовой |
| SettlementID | Числовой |
| ContractName | Текстовый |
| DateStart | Дата |
| DateStop | Дата |
| Cost | Денежный |

Таблица 7 – Поля таблицы «Prolonging»

| Название поля | Тип данных |
| --- | --- |
| ProlongingID | Числовой |
| TreatyID | Числовой |
| Prolong | Дата |
| Comment | Текстовый |

Таблица 8 – Поля таблицы «Capture»

| Название поля | Тип данных |
| --- | --- |
| ActionID | Числовой |
| PatrolID | Числовой |
| Chief | Текстовый |
| Brand | Текстовый |
| Document | Текстовый |

Таблица 9 – Поля таблицы «Calling»

| Название поля | Тип данных |
| --- | --- |
| CallingID | Числовой |
| TreatyID | Числовой |
| DateTime | Дата |
| False | Логический |
| TaxID | Числовой |
| CompensationID | Числовой |

Таблица 10 – Поля таблицы «User»

| Название поля | Тип данных |
| --- | --- |
| Registr | Числовой |
| Логин | Текстовый |
| Пароль | Текстовый |
| Права доступа | Текстовый |

После того, как таблицы созданы, необходимо создать между ними связи.

В приложении «А» На рисунке 3 представлена физическая модель данных, отражающая все таблицы и их поля, а также связи между таблицами.

## 4.2. Разработка программных модулей информационной системы «Охрана квартир»

Модульное программирование — это организация программы как совокупности небольших блоков, называемых модулями, структура и поведение которых подчиняются определённым правилам. Использование модульного программирования позволяет упростить тестирование программы и обнаружение ошибок. Аппаратно-зависимые подзадачи могут быть строго отделены от других подзадач, что улучшает мобильность создаваемых программ.

Принцип модульности является средством упрощения задачи проектирования программного обеспечения и распределения процесса разработки. При разбиении ПО на модули для каждого модуля указывается реализуемая им функциональность, а также связи с другими модулями. Удобство использования модульной архитектуры заключается в возможности обновления или замены модуля, без необходимости изменения остальной системы.

Разработанные в MS Visual Studio формы продемонстрированы в

**Вывод по главе.**

# 5 РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ

Тестирование программного обеспечения – процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.

Тестовый сценарий – это неавтоматизированный или автоматизированный сценарий, содержащий инструкции по реализации тестового набора. Тестовый сценарий может быть написан вручную (для выполнения человеком) либо полностью или частично автоматизирован.

## 5.1 Назначение эксперимента. Выбор и обоснование методики проведения тестирования

Можно определить такие основные цели тестирования программного обеспечения:

* Предоставление информации о качестве ПО конечному заказчику.
* Повышение качества ПО.
* Предотвращение появления дефектов.

Цели тестирования могут отличаться, в зависимости от этапа разработки программного обеспечения, на котором оно проводится. К примеру, на этапе кодирования целью тестирования будет вызов как можно большего количества сбоев в работе программы, что позволит локализовать и исправить дефекты. В то же время, при приемочном тестировании необходимо показать, что система работает правильно. В период сопровождения, тестирование в основном необходимо для того, чтобы убедится в отсутствии новых багов, появившихся во время внесения изменений.

Главная же задача тестирования – поиск дефектов.

## 5.2 Технология тестирования

## 5.3 Результаты проведения тестирования

# ЗАКЛЮЧЕНИЕ

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. IT-GOST.RU [Электронный ресурс]: Теория и практика UML. Диаграмма деятельности, 2023. URL: <http://it-gost.ru/articles/view_articles/96> (дата обращения: 20.11.2023).
2. [Электронный ресурс]: Программа профессионального модуля ПМ 01. Разработка модулей программного обеспечения для компьютерных систем,2019. URL: <https://cs.гапоу-кгпт.рф/-/dpBw6QvcMrs97SUYUOCWug/sv/document/e2/66/37/682986/2747/РП%20ПМ%2001.pdf?1680595748> (дата обращения: 18.11.2023).

# ПРИЛОЖЕНИЕ A



Рисунок 1 - Набор данных к информационной системе «Охрана квартир»

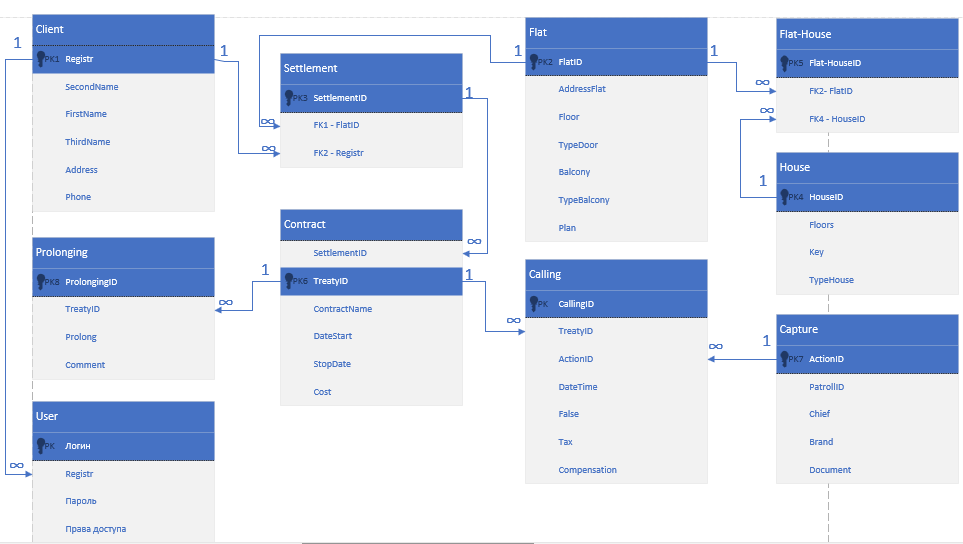


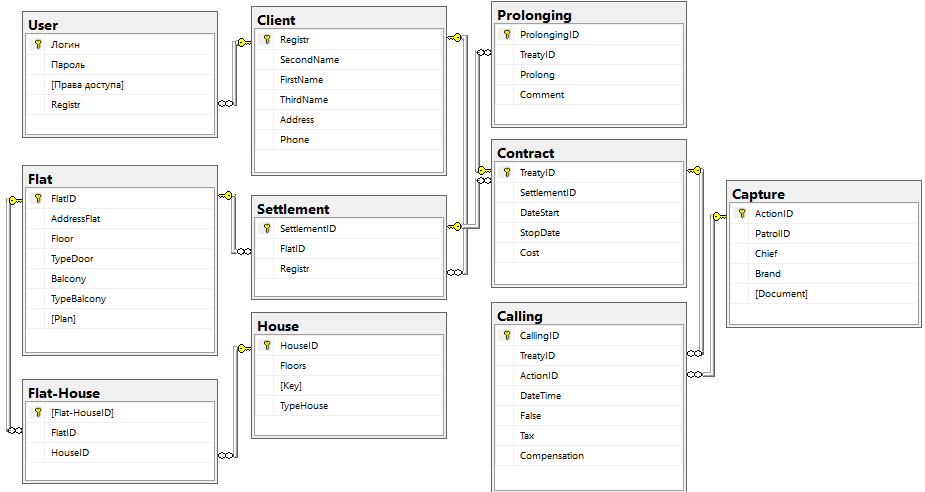
Рисунок 2 – Логическая модель данных 

Рисунок 3 – Физическая модель данных

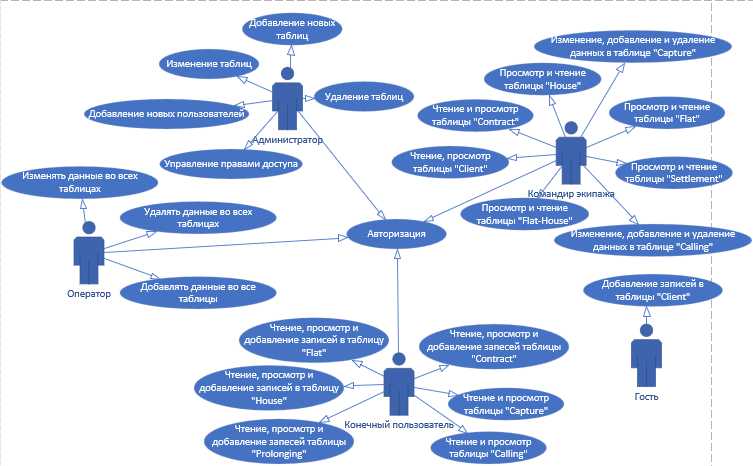


Рисунок 4 – Диаграмма вариантов использования

# ПРИЛОЖЕНИЕ Б