Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  
«Национальный исследовательский университет  
«Высшая школа экономики»

Вечерне-заочный факультет экономики и управления

**Одинцов Александр Сергеевич**

**РАЗРАБОТКА СИСТЕМЫ КОММУНИКАЦИИ ДЛЯ УПРАВЛЕНИЯ МНОГОКВАРТИРНЫМ ДОМОМ**

*Выпускная квалификационная работа*

**по направлению подготовки 09.03.04 Программная инженерия**

**образовательная программа бакалавриата «Программная инженерия»**

Руководитель

старший преподаватель кафедры информационных технологий в бизнесе

В.В. Ланин

г. Пермь, 2021 год

Оглавление

[Сбор и анализ требований 7](#_Toc88757967)

[Бизнес-идея 7](#_Toc88757968)

[Видение 7](#_Toc88757969)

[Определение целевой аудитории системы 7](#_Toc88757970)

[Контекстные сценарии 10](#_Toc88757971)

[Контекстный сценарий для жильца многоквартирного дома 10](#_Toc88757972)

[Контекстный сценарий для собственника коммерческого помещения 11](#_Toc88757973)

[Контекстный сценарий для управляющего многоквартирным домом 13](#_Toc88757974)

[Анализ конкурентов 13](#_Toc88757975)

[Предварительный документ о требованиях 15](#_Toc88757976)

[Функциональные требования 18](#_Toc88757977)

[Диаграммы прецедентов 19](#_Toc88757978)

[Логическая модель системы 22](#_Toc88757979)

[Матрица создания, чтения, модификации и удаления данных 23](#_Toc88757980)

[Словарь данных 24](#_Toc88757981)

[База данных системы 30](#_Toc88757982)

[Проектирование интерфейса системы 34](#_Toc88757983)

[Разработка системы 39](#_Toc88757984)

[Выбор инструментальных средств разработки 39](#_Toc88757985)

[Написание моделей веб-приложения 40](#_Toc88757986)

[Заключение 43](#_Toc88757987)

Аннотация

Выпускная квалификационная работа на тему «Разработка системы коммуникации для управления многоквартирным домом»

**Автор:** Одинцов Александр Сергеевич, гр. ПИ-18У

**Руководитель:** Ланин Вячеслав Владимирович, старший преподаватель

В рамках выпускной квалификационной работы производится постановка бизнес-идеи, определение пользовательских требований, формирование функциональных требований, техническое проектирование и разработка системы коммуникации для управления многоквартирным домом.

Приводятся контекстные сценарии использования системы, документ о требованиях, диаграммы прецедентов и последовательностей разрабатываемого приложения, рассматриваются алгоритмы, описываются классы и интерфейс пользователя.

Работа состоит из <?> страниц, содержит <?> литературных источников, <?> рисунков и <?> таблиц.

Введение

Современное общество находится в стадии активной информатизации и компьютеризации процессов, затрагивающих различные сферы жизни – как повседневной, так и деловой. На сегодняшний день практически любая деятельность человека основывается на использовании информационных технологий: образование от дошкольного до высшего, производство, сферы обслуживания и управления и пр. Наравне с различными сферами жизни общества, в отрасли жилищно-коммунального хозяйства, хоть и с некоторым отставанием, также идёт внедрение современных информационных технологий [1]. Рост использования электронных сервисов в области ЖКХ становится важной составляющей в процессе перехода к современному информационному обществу. Это позволяет организовать более прозрачный и эффективный контроль за работой служб ЖКХ, внедрить современные цифровые сервисы для граждан, организовать удобную работу с обращениями собственников жилых и коммерческих помещений.

Нынешняя тенденция такова, что всё большее количество граждан выражает желание иметь возможность удобной и оперативной коммуникации с управляющими компаниями, поставщиками коммунальных услуг и ресурсов, а также руководством ассоциаций домовладельцев и главами муниципалитетов. Это обуславливается тем фактом, что собственники жилых и коммерческих помещений недовольны отношением работников жилищно-коммунального хозяйства к своим непосредственным обязанностям [2].

Главная задача участников системы ЖКХ — это обеспечение безопасного и комфортного проживания и использования помещений, поэтому внедрение современных технологий, которые способствуют формированию удобной среды, на руку всем [1]. Таким образом, каждой из заинтересованных сторон цифровизация жилищно-коммунального хозяйства будет по-своему выгодна.

В первую очередь, необходимо выяснить, что подразумевает понятие «Управление многоквартирным домом». Согласно Жилищному кодексу Российской Федерации, это согласованная деятельность собственников жилья по созданию и поддержанию благоприятных и безопасных условий проживания в доме, которая обеспечивает содержание общедомового имущества надлежащим образом, решение возникающих вопросов использования данного имущества, и, кроме того, предоставление коммунальных услуг гражданам, которые проживают в доме [3].

Современные информационные технологии могут позволить повысить качество принимаемых решений, социальную защищенность населения и усилить контроль за жилищно-коммунальной сферой деятельности [4].

Одной из основных проблем в ЖКХ на сегодняшний день зачастую является некоторая апатия жильцов многоквартирного дома в части управления общедомовым имуществом. Цифровые технологии способны помочь преодолеть этот барьер. Общие чаты позволяют вовлекать собственников в процесс управления общедомовым имуществом. Кроме того, иные формы взаимодействия посредством цифровых технологий, такие как электронные формы обращений и онлайн-голосования, упрощают взаимодействие граждан и управляющей организации, увеличивают степень участия жильцов в вопросах улучшения качества услуг ЖКХ [1].

Сегодня для решения данных задач в основном используются общедомовые чаты, созданные в различных мессенджерах (Telegram, Viber, WhatsApp, а также группы в социальной сети ВКонтакте). Такой способ коммуникации обладает множеством недостатков: сообщения идут потоком, много спама, обсуждение отвлечённых тем, при этом важные вопросы упускаются или остаются незамеченными, отсутствует взаимодействие с представителями управляющей организации; в чаты и группы могут попасть посторонние лица, не имеющие отношения к собственникам.

В разрабатываемой в рамках квалификационной работы системе коммуникации предполагается более тесное и оперативное взаимодействие между стейкхолдерами: система позволит более организованно подойти к вопросу управления путём регистрации обращений и получения обратной связи от управляющей организации, а также обмена информацией между собственниками помещений.

**Объектом** квалификационной работы является автоматизация коммуникации между участниками процесса управления многоквартирным домом.

**Предмет** квалификационной работы – система коммуникации и регистрации обращений граждан в управляющую организацию.

**Целью** данной квалификационной работы является разработка системы обращений, служащей для оптимизации процесса управления многоквартирным жилым домом.

В рамках квалификационной работы предполагается выполнение следующих **задач**:

1. Постановка бизнес-идеи и определение целевой аудитории системы.
2. Подготовка документа о требованиях к системе.
3. Проектирование системы.
4. Выбор инструментов разработки и технологических решений, подходящих для решения поставленной задачи.
5. Разработка и тестирование системы.
6. Подготовка документации пользователя.

# Сбор и анализ требований

## Бизнес-идея

Для любой разрабатываемой системы, в первую очередь, необходимо определить её бизнес-идею. В нашем случае, система коммуникации для управления многоквартирным домом должна обеспечивать оперативность функций коммуникации и контроля в управлении многоквартирным домом для жильцов и собственников помещений с одной стороны, и для управляющей организации (УК или ТСЖ) с другой. Коммуникация осуществляются посредством текстовых заявок, которые, помимо самого текста заявки, должны иметь служебную информацию, необходимую для контроля исполнения.

## Видение

Система коммуникации для управления многоквартирным домом создаётся для жильцов и собственников помещений, которым необходимо средство, служащее для коммуникации с управляющей организацией, в целях более оперативного решения вопросов по управлению домом, по сравнению с традиционными средствами.

Система так же необходима и управляющей организации, для более эффективного и оперативного решения вопросов и проблем жильцов многоквартирного дома и собственников помещений.

Преимуществом системы над традиционными средствами коммуникации является оперативность постановки задач и прозрачность их исполнения. Также, система не требует от пользователей специфических знаний в области информационных технологий и интеграции с государственными системами.

## Определение целевой аудитории системы

Перед описанием контекстных сценариев для различных типов пользователей системы, необходимо определить эти типы. Для этого рассмотрим примеры нескольких пользователей системы, каждый из которых будет представлять собой определённый тип.

Таблица 1. Целевая аудитория системы, пользователь №1

|  |  |
| --- | --- |
| Имя | Александр Сергеевич Пушкин |
| Деятельность или род деятельности | Жилец многоквартирного дома |
| Возраст | 37 |
| Образование | Высшее |
| Отношение к инновациям | Консервативный |
| Потребительский стаж | Продвинутый |
| Тип | Пользователь |
| Роль в семье | Глава семьи |
| Описание |  |
| О чем болит голова | Невозможность проконтролировать работу УК без постоянных телефонных звонков управляющему. Заявки, поданные голосом, часто «забываются» |
| Продукты-заменители, которые он использует | Голосовая коммуникация (телефон) |
| Ожидание от продукта | Лёгкость в освоении, контроль исполнения заявок |
| Что читает онлайн, офлайн | Федеральные и региональные новости (РИА, Лента.ру, 59.ру), авто новости, спортивные новости |
| Уровень доходов | 105 000 |
| Что мотивирует потребителя |  |
| Девайс | Смартфон |
| Предпочитаемый формат | Веб-приложение |

Таблица 2. Целевая аудитория системы, пользователь №2

|  |  |
| --- | --- |
| Имя | Михаил Юрьевич Лермонтов |
| Деятельность или род деятельности | Собственник коммерческого помещения в многоквартирном доме |
| Возраст | 26 |
| Образование | Высшее неоконченное |
| Отношение к инновациям | Раннее большинство |
| Потребительский стаж | Продвинутый |
| Тип | Пользователь |
| Роль в семье | Холост |
| Описание |  |
| О чем болит голова | Поданные в письменном виде заявления на имя председателя ТСЖ очень медленно обрабатываются. Телефонные звонки председателю возможны только дважды в неделю в часы приёма |
| Продукты-заменители, которые он использует | Голосовая коммуникация (телефон). Письменные обращения |
| Ожидание от продукта | Оперативность коммуникации |
| Что читает онлайн, офлайн | Новости из мира финансов (РБК, Форбс) |
| Уровень доходов | 88 000 |
| Что мотивирует потребителя |  |
| Девайс | Ноутбук |
| Предпочитаемый формат | Веб-приложение |

Таблица 3. Целевая аудитория системы, пользователь №3

|  |  |
| --- | --- |
| Имя | Николай Васильевич Гоголь |
| Деятельность или род деятельности | Управляющий многоквартирным домом |
| Возраст | 42 |
| Образование | Среднее-специальное |
| Отношение к инновациям | Позднее большинство |
| Потребительский стаж | Эксперт |
| Тип | Заказчик, пользователь |
| Роль в семье | Муж, отец |
| Описание |  |
| О чем болит голова | Не хватает организации поступающих заявок, найм секретаря не предусмотрен бюджетом. |
| Продукты-заменители, которые он использует | Голосовая коммуникация (телефон). Письменные обращения |
| Ожидание от продукта | Возможность оперативного получения информации о возникающих у жильцов проблемах для более эффективного их решения |
| Что читает онлайн, офлайн | Новости из мира техники, классическую литературу |
| Уровень доходов | 93 000 |
| Что мотивирует потребителя |  |
| Девайс | Стационарный компьютер |
| Предпочитаемый формат | Веб-приложение |

## Контекстные сценарии

Для каждого из типов пользователей определяется свой контекстный сценарий, исходя из предполагаемого использования системы.

### Контекстный сценарий для жильца многоквартирного дома

1. Александр Сергеевич Пушкин проживает на 20 этаже элитного многоквартирного дома. Обычные средства коммуникации с УК недостаточно удобны: заявки, поданные по телефонной связи, необходимо многократно дублировать, по ним нет обратной связи. Бывали случаи, когда поданная голосом заявка игнорировалась. Вместо этого, Александр Сергеевич предпочитает использовать систему коммуникации для управления многоквартирным домом.
2. Одна из кнопок вызова лифта на 20 этаже дома не работает. Чтобы подать заявку, Александр Сергеевич входит в систему, используя выданный УК логин и пароль.
3. Первое, что он видит после входа в систему – список ранее поданных им обращений. Незакрытые обращения отображаются наверху списка.
4. Александр Сергеевич нажимает кнопку «Новая заявка».
5. Он выбирает категорию «Лифт» и в подробностях описывает возникшую проблему.
6. Далее Александр Сергеевич нажимает кнопку «Подать заявку», после чего форма подачи заявки закрывается. Вновь поданная заявка появляется наверху списка заявок.
7. На основании категории заявки, ей автоматически присваивается приоритет «высокий».
8. Представитель УК получает сообщение о поступившей заявке. При входе представителя УК в систему, он видит вновь поданную заявку наверху списка поданных заявок. На основе приоритета, заявка имеет SLA 1 час. Представитель УК самостоятельно связывается с организацией, обслуживающей лифты в доме, и контролирует выполнение ремонта. Александр Сергеевич никак не контактирует с лифтёром.
9. После успешно выполненного ремонта, Александр Сергеевич получает уведомление о том, что его заявка была выполнена. В списке заявок на стороне Александра Сергеевича выполненная заявка отображается ниже активных.

### Контекстный сценарий для собственника коммерческого помещения

1. Михаил Юрьевич Лермонтов является собственником коммерческого помещения на первом этаже многоквартирного дома. Наземная парковка перед домом является частью общедомовой собственности, но из-за отсутствия шлагбаума она всегда полностью занята посторонними автомобилями, принадлежащими жильцам соседних домов. Михаил Юрьевич хотел бы поднять вопрос об ограничении доступа посторонних автомобилей на парковку посредством установки шлагбаума. Без системы коммуникации для управления многоквартирным домом, данный вопрос можно было бы поднять только на ежеквартальном заседании совета жильцов дома. Используя систему, Михаил Юрьевич может инициировать внеочередное обсуждение установки шлагбаума.
2. Михаил Юрьевич входит в систему, используя свои учётные данные.
3. Для того, чтобы вынести вопрос на обсуждение Михаил Юрьевич нажимает кнопку «Новая заявка».
4. Он выбирает тип «Голосование», описывает аргументы в пользу установки шлагбаума в текстовом окне, прикладывает два варианта ответа – «За» и «Против».
5. Далее он отмечает пункт «Публикация предварительных результатов», чтобы текущее количество голосов за каждый из вариантов было видно в реальном времени, до завершения голосования.
6. После этого, Михаил Юрьевич нажимает кнопку «Подать заявку». Форма закрывается, и система возвращает пользователя в список заявок.
7. На основании категории заявки, она появляется на панели объявлений у всех зарегистрированных в рамках данного многоквартирного дома пользователей.
8. Пользователи, являющиеся собственниками квартир и помещений, имеют возможность проголосовать в заявке и принять участие в обсуждении.
9. Представитель УК принимает дальнейшее решение в зависимости от результата голосования:
   1. Большинство «За» − заявка принимается в работу, установка шлагбаума закладывается в бюджет. SLA по заявке увеличивается, представитель УК оставляет комментарии по текущему статусу работ по установке внутри заявки.
   2. Большинство «Против» − представитель УК закрывает заявку без дальнейшего движения.
   3. Голосов примерно поровну, плюс минус 10% от общего количества жильцов − заявка устанавливается в статус «Приостановлена». Вопрос об установке шлагбаума будет выноситься на очный совет жильцов дома.

### Контекстный сценарий для управляющего многоквартирным домом

1. Николай Васильевич является управляющим многоквартирным домом. Он размещает объявление о необходимости согласовать дату и время допуска в жилые помещения специалистов для проверки вентиляции. Летом, в период отпусков, многие жильцы дома уезжают на отдых, поэтому не все имеют возможность ознакомиться с объявлением, размещённым на доске в холле дома. На обзвон каждого жильца затрачивается много времени.
2. Николай Васильевич входит в систему, используя свои логин и пароль.
3. Он нажимает «Новая заявка» и выбирает тип «Объявление».
4. В форме заявки Николай Васильевич выбирает категорию «Объявления от Управляющей компании» заполняет текст объявления с просьбой к жильца указать удобные даты и время, в которые они будут находиться дома и смогут принять специалистов. После этого Николай Васильевич нажимает «Подать заявку».
5. Исходя из категории заявки «Объявление», комментарии жильцов, оставленные внутри заявки, будут видны только автору заявки. Жильцы не будут видеть комментарии друг друга.
6. Получив необходимую информацию, Николай Васильевич может составить поквартирный план проверки вентиляции.

## Анализ конкурентов

Для обоснования актуальности разработки системы, необходимо проанализировать решения для управления многоквартирным домом, которые используются на текущий момент.

В большинстве случаев для коммуникации жильцов с управляющей организацией используются такие решения как телефонная коммуникация и письменные обращения. Как мы уже смогли убедиться на примере контекстных сценариев, данные средства во многих случаях являются недостаточно эффективными.

Телефонная коммуникация оперативна, но не всегда доступна. Представитель управляющей организации может оказаться занят и не сможет вовремя ответить на звонок. Во время форс-мажорных ситуаций, например при аварийном отключении водоснабжения, управляющий домом будет тратить большое количество времени на многочисленные поступающие звонки. Аналогично, при необходимости запросить информацию от всех жильцов дома, необходимо провести общий обзвон, что довольно трудоёмко.

Письменная коммуникация является очень медленным способом донесения информации. При размещении объявления на физической доске, оно недостаточно оперативно принимается во внимание жильцами дома, особенно в период отпусков.

Жильцы и собственники, которые хотят подать заявку в управляющую организацию в письменном виде вынуждены ждать, пока заявка будет принята на рассмотрение. Ответа, в случае если он также будет в письменном виде, придётся ожидать долго.

Одной из попыток оптимизировать коммуникацию между собственниками жилых помещений и управляющей организацией является система ГИС ЖКХ, операторами которой являются АО «Почта России» и «Ланит». Эта система предлагает следующий набор возможностей [<https://giszhkh.ru/>]:

1. Оплата счетов за услуги ЖКХ;
2. Ввод и проверка показаний приборов учета;
3. Контроль за работами по дому, проводимыми управляющими компаниями;
4. Направление обращений и жалоб в органы власти;
5. Принятие участия в совместных электронных голосованиях и обсуждение вопросов и проблем с соседями на форуме.

Основным предназначением системы ГИС ЖКХ является интеграция с городскими службами и объединение нескольких функций в рамках единого приложения. Такие действия как оплата счетов за услуги ЖКХ, ввод показаний приборов учёта и направление обращений в органы власти уже возможно осуществлять в электронном виде, используя веб-сайты соответствующих ведомств, без необходимости использовать систему ГИС ЖКХ.

В то же время для использования любых возможностей этой системы нужно иметь подтверждённую учётную запись портала Госуслуг. Она необходима даже для внутридомового использования – коммуникации с управляющей организацией и другими жильцами, что может быть принципиально неприемлемо для некоторых пользователей.

Система коммуникации для управления многоквартирным домом не требует взаимодействия с порталом Госуслуг. Регистрация пользователей в системе производится управляющей компанией или ТСЖ, без привлечения третьих лиц и организаций и без необходимости обязывать жильцов заводить какие-либо аккаунты самостоятельно. Таким образом можно привлечь к совместному управлению домом наибольшее количество собственников.

## Предварительный документ о требованиях

Документ о требованиях служит для того, чтобы все требования к системе были учтены, и никакая важная информация не была упущена. Требования поясняют, что, с точки зрения пользователей, должна выполнять система для решения поставленных перед ней задач. Так же учитываются пожелания пользователей, то есть, что они хотят видеть в конечном продукте.

Таблица 4. Предварительные требования к системе

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Идентификатор | Описание требования | Приоритет | Источник | Дополнительные вопросы |
| **B1** | Создание и контроль заявок в управляющую организацию | Высокий | Собственник помещения |  |
| *U1.1* | Возможность создать заявку | Высокий | Собственник помещения | Сколько символов может быть в тексте заявки? |
| *U1.2* | Возможность выбрать категорию заявки | Высокий | Собственник помещения | Какие категории можно выбрать? Как категории влияют на приоритет и SLA заявки? |
| *U1.3* | Возможность отправить созданную заявку на исполнение представителю управляющей организации | Высокий | Собственник помещения |  |
| *U1.4* | Возможность видеть текущий статус своей заявки | Высокий | Собственник помещения | Какие статусы у заявок должны быть? |
| *U1.5* | Возможность прикрепить фото к создаваемой заявке | Средний | Собственник помещения | Какие ограничения по фото? (количество и размер) |
| *U1.6* | Получение уведомлений по обновлению статуса заявки | Средний | Собственник помещения | В каком виде должны поступать уведомления? |
| **B2** | Получение о обработка заявок от собственников помещений | Высокий | Представитель управляющей организации |  |
| *U2.1* | Возможность получать заявки | Высокий | Представитель управляющей организации |  |
| *U2.2* | Возможность получать уведомления о новых заявках | Средний | Представитель управляющей организации | В каком виде должны поступать уведомления? |
| *U2.3* | Возможность изменять статусы заявок | Высокий | Представитель управляющей организации | Какие статусы у заявок должны быть? |
| *U2.4* | Возможность сортировать заявки для более удобной работы с ними | Низкий | Представитель управляющей организации | По каким критериям должна производиться сортировка? |
| *U2.5* | Возможность пре-модерации для заявок пользователей, злоупотребляющих системой | Низкий | Представитель управляющей организации | Каким образом? По каким критериям должна проводиться пре-модерация? |
| **B3** | Общие объявления | Высокий | Представитель управляющей организации и собственник помещения |  |
| *U3.1* | Возможность размещать объявления, доступные всем пользователям системы | Высокий | Представитель управляющей организации и собственник помещения | Сколько символов может быть в тексте объявления? |
| *U3.2* | Возможность удалять свои объявления | Высокий | Представитель управляющей организации и собственник помещения | Возможно ли задать автоматическое удаление объявления по достижению даты/времени? |
| *U3.3* | Возможность прикреплять фото к создаваемым объявлениям | Средний | Представитель управляющей организации и собственник помещения | Какие ограничения по фото? (количество и размер) |
| *U3.4* | Возможность снимать объявления с публикации | Средний | Представитель управляющей организации и собственник помещения |  |
| *U3.5* | Возможность задать дату автоматического снятия объявления с публикации | Средний | Представитель управляющей организации и собственник помещения |  |
| *U3.6* | Возможность выбрать категорию объявления | Средний | Представитель управляющей организации и собственник помещения |  |
| **B4** | Электронное голосование | Высокий | Представитель управляющей организации и собственник помещения |  |
| *U4.1* | Возможность размещать электронное голосование, доступное всем пользователям системы | Высокий | Представитель управляющей организации и собственник помещения | Сколько пунктов голосования может быть? Возможен ли пункт «свой вариант»? Можно ли устанавливать временные ограничения по голосованию? |
| *U4.2* | Возможность принимать участие в электронном голосовании | Высокий | Собственник помещения | Возможно ли изменить свой голос? |
| *U4.3* | Возможность выбирать тип голосования – открытое или закрытое | Средний | Представитель управляющей организации и собственник помещения |  |
| *U4.4* | Возможность видеть предварительные результаты голосования | Средний | Представитель управляющей организации и собственник помещения | Логическая переменная? |

## Функциональные требования

1. Система должна предоставлять пользователям возможность создать заявку с указанием её заголовка и текста и прикрепить изображение.
2. Должна присутствовать поддержка трёх типов заявок – заявка на исполнение, голосование, объявление.
3. Заявки на исполнение должны поддерживать возможность выбрать категорию при создании, от которой будет зависеть её приоритет.
4. В заявках на исполнение должна быть возможность изменить статус и исполнителя.
5. Статус заявки должен отражать текущий этап жизненного цикла – новая заявка, заявка в работе, приостановленная заявка, завершённая заявка, отклонённая заявка.
6. В голосованиях должна быть возможность указать варианты голосования, их описание и срок завершения при создании.
7. Должна быть возможность проводить как открытое, так и закрытое голосование.
8. Должна быть возможность выбрать отображение предварительных результатов голосования при его создании.
9. В голосованиях должна быть возможность оставить голос для любого пользователя-собственника. Вес голоса должен равняться количеству помещений в собственности пользователя.
10. При создании объявления должна быть возможность выбрать его категорию.
11. В объявлениях должна быть возможность указать срок актуальности, после которого объявление снимается с публикации.
12. Должна быть возможность снять объявление с публикации вручную.
13. Пользователь-собственник должен иметь возможность самостоятельно изменить часть своих учётных данных – пароль, телефон, e-mail, изображение профиля.
14. Пользователь-представитель управляющей организации должен иметь возможность создавать учётные записи пользователей, с указанием их учётных данных и устанавливать связь между пользователями-собственниками и помещениями дома.
15. Пользователь-представитель управляющей организации должен иметь возможность создавать новые объекты класса «помещение» в системе.

## Диаграммы прецедентов

Ниже представлены диаграммы прецедентов разрабатываемой системы. На диаграмме прецедентов подсистемы работы с заявками отражены возможные действия пользователей классов «собственник» и «представитель управляющей организации» по отношению к формированию и обработке заявок, включая заявки-голосования и объявления.

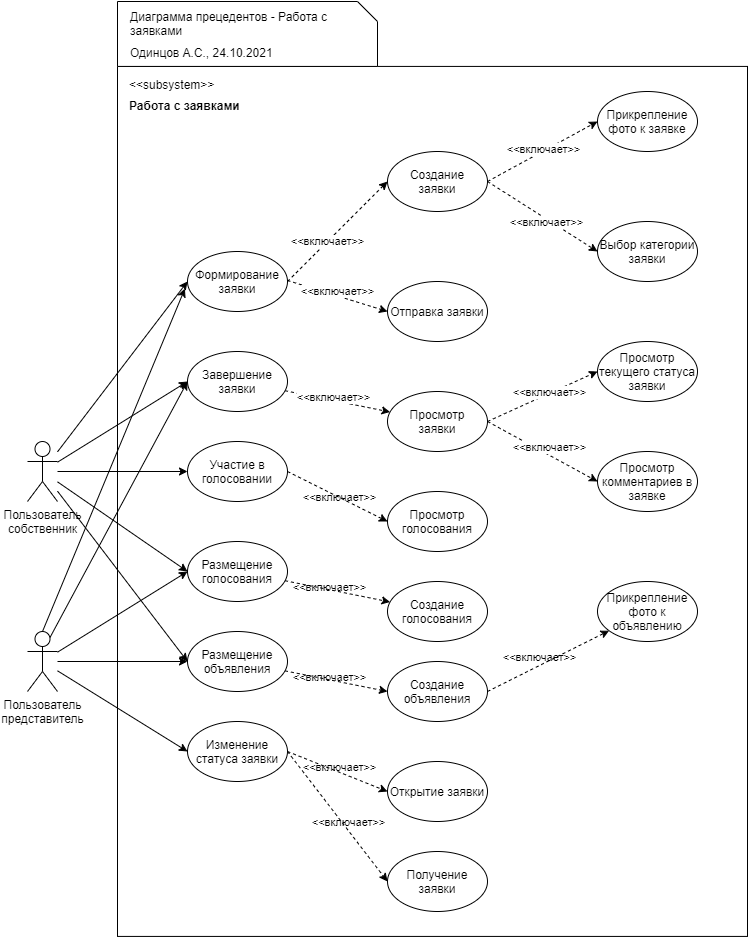


Рис 1. Диаграмма прецедентов работы с заявками

Диаграмма прецедентов работы с учётными записями пользователей рассматривает возможные действия пользователей по авторизации в системе и изменению своих учётных данных, а также дополнительные возможности пользователей класса «представитель управляющей организации» по созданию и блокировке учётных записей и заведению новых объектов класса «помещение» в систему.

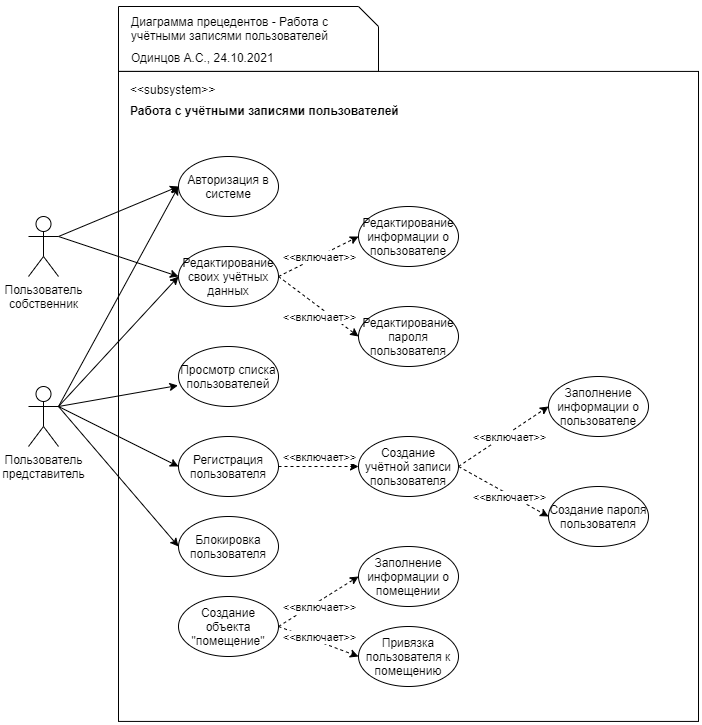


Рис 2. Диаграмма прецедентов работы с учётными записями

На рисунке 3 изображён жизненный цикл заявки в системе, начиная от её создания до завершения, в том числе при отклонении заявки.

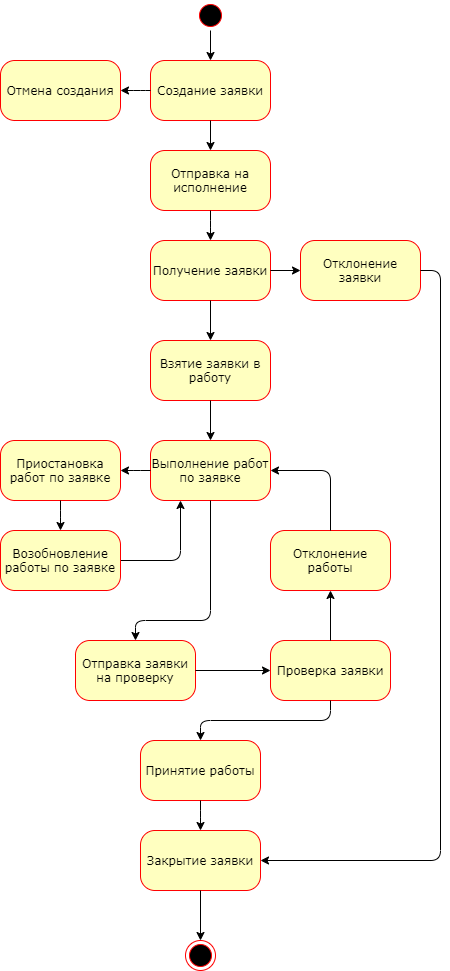


Рис 3. Диаграмма состояний заявки в системе

## Логическая модель системы

После анализа предметной области, на основе диаграмм прецедентов создана диаграмма бизнес-классов, которая определяет сущности внутри системы и взаимосвязь между ними.

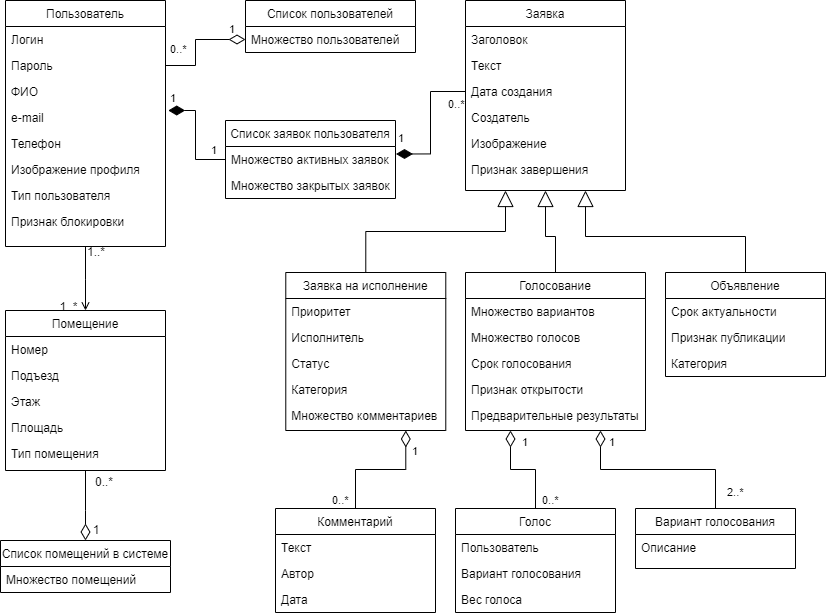


Рис 4. Диаграмма бизнес-классов системы

## Матрица создания, чтения, модификации и удаления данных

На основе имеющегося списка классов системы создаётся матрица CRUD, которая позволяет определить соответствие пользовательских прецедентов операциям создания, чтения, модификации и удаления данных по отношению к каждому из классов.

Таблица 5. CRUD-матрица системы

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Прецедент** | **Список пользователей** | **УЗ пользователя** | **Список помещений** | **Помещение** | **Список заявок** | **Заявка** | **Голосование** | **Объявление** | **Комментарий** | **Голос** | **Вариант голосования** |
| Создание заявки |  |  |  |  | U | C |  |  |  |  |  |
| Прикрепление фото к заявке |  |  |  |  |  | U |  |  |  |  |  |
| Выбор категории заявки |  |  |  |  |  | U |  |  |  |  |  |
| Отправка заявки |  |  |  |  |  | U |  |  |  |  |  |
| Просмотр заявки |  |  |  |  | R | R |  |  |  |  |  |
| Просмотр статуса заявки |  |  |  |  |  | R |  |  |  |  |  |
| Просмотр комментариев заявки |  |  |  |  |  | R |  |  | R |  |  |
| Изменение статуса заявки |  |  |  |  |  | U |  |  |  |  |  |
| Завершение заявки |  |  |  |  |  | U |  |  |  |  |  |
| Создание голосования |  |  |  |  |  |  | C |  |  |  | C |
| Размещение голосования |  |  |  |  |  |  | U |  |  |  |  |
| Просмотр голосования |  |  |  |  |  |  | R |  |  |  | R |
| Участие в голосовании |  |  |  |  |  |  |  |  |  | C | R |
| Создание объявления |  |  |  |  |  |  |  | C |  |  |  |
| Прикрепление фото к объявлению |  |  |  |  |  |  |  | U |  |  |  |
| Размещение объявления |  |  |  |  |  |  |  | U |  |  |  |
| Авторизация в системе | R | R |  |  |  |  |  |  |  |  |  |
| Редактирование учётных данных |  | U |  |  |  |  |  |  |  |  |  |
| Просмотр списка пользователей | R |  |  |  |  |  |  |  |  |  |  |
| Создание УЗ пользователя |  | C |  |  |  |  |  |  |  |  |  |
| Блокировка УЗ пользователя |  | U |  |  |  |  |  |  |  |  |  |
| Создание объекта «помещение» |  |  | U | C |  |  |  |  |  |  |  |
| Редактирование информации о помещении |  |  |  | U |  |  |  |  |  |  |  |
| Привязка пользователя к помещению |  | R |  | U |  |  |  |  |  |  |  |

## Словарь данных

Словарь данных разрабатываемой системы описывается в целях определения структур данных. Каждой из них назначается свои формат и длина данных, а также диапазон значений.

Таблица 6. Словарь данных

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Элемент данных** | **Значение по умолчанию** | **Описание** | **Структура или тип данных** | **Длина** | **Значение** |
| **Заявка** |  |  | Заголовок  Идентификатор заявки  Текст  Дата создания  Идентификатор создателя  Изображение  Признак завершения |  |  |
| Заголовок | «Новая заявка» | Текстовый заголовок заявки |  | 256 | Текстовая строка |
| Идентификатор заявки | - | Уникальный идентификатор объекта класса «заявка» |  | 8 | Автоматически генерируемый номер |
| Текст | - | Текст заявки |  | 8192 | Текстовая строка |
| Дата создания | Текущая дата | Дата создания заявки |  | 16 | YYYY-MM-DD hh:mm |
| Создатель | Текущий пользователь | Идентификатор, ссылающийся на пользователя |  | 8 | Автоматически генерируемый номер |
| Изображение | - | Файл изображения |  | - | Графический файл |
| Признак завершения | false | Признак того, что заявка завершена |  | 1 | Логический тип |
| **Заявка на исполнение** |  |  | Приоритет  Идентификатор исполнителя  Статус  Категория  Множество комментариев |  |  |
| Приоритет | 1 («Средний») | Важность исполнения заявки для сортировки |  | 1 | Число от 0 до 4 |
| Идентификатор исполнителя | - | Идентификатор, ссылающийся на пользователя |  |  | Автоматически генерируемый номер |
| Статус | 0 («Новая») | Текущий этап работ по заявке |  | 1 |  |
| Категория | 0 («Общая») | К какой сфере работ относится заявка |  | 1 |  |
| Множество комментариев | - | Структура, содержащая комментарии исполнителей заявки |  |  |  |
| **Голосование** |  |  | Множество вариантов  Множество голосов  Срок голосования  Предварительные результаты  Открытое |  |  |
| Множество вариантов | - | Структура, содержащая варианты голосования |  |  |  |
| Множество голосов | - | Структура, содержащая голоса пользователей |  |  |  |
| Срок голосования | Текущая дата + 7 дней | Срок, когда голосование завершается |  | 10 | YYYY-MM-DD hh:mm |
| Предварительные результаты | false | Признак того, что результаты голосования будут видны до его завершения |  | 1 | Логический тип |
| Открытое | false | Признак того, что голосование является открытым |  | 1 | Логический тип |
| **Объявление** |  |  | Срок актуальности  Категория  Опубликовано |  |  |
| Срок актуальности | Текущая дата + 7 дней | Срок, когда объявление теряет актуальность и снимается с публикации |  | 10 | YYYY-MM-DD |
| Категория | 0 («Общее») | Категория объявления |  |  |  |
| Опубликовано | false | Признак того, что объявление опубликовано |  | 1 | Логический тип |
| Категория | 0 («Общее») | Категория объявления |  |  |  |
| **Комментарий** |  |  | Идентификатор комментария  Текст  Автор  Дата |  |  |
| Идентификатор комментария | - | Уникальный идентификатор объекта класса «комментарий» |  | 8 | Автоматически генерируемый номер |
| Текст | - | Текст комментария |  | 4096 | Текстовая строка |
| Автор | Текущий пользователь | Идентификатор, ссылающийся на пользователя |  | 8 | Автоматически генерируемый номер |
| Дата | Текущая дата | Дата добавления комментария |  | 10 | YYYY-MM-DD hh:mm |
| **Голос** |  |  | Идентификатор голоса  Пользователь  Вариант голосования  Вес голоса |  |  |
| Идентификатор голоса | - | Уникальный идентификатор объекта класса «голос» |  | 8 | Автоматически генерируемый номер |
| Пользователь | Текущий пользователь | Идентификатор, ссылающийся на пользователя |  | 8 | Автоматически генерируемый номер |
| Вариант голосования | - | Идентификатор, ссылающийся на вариант голосования |  | 8 | Автоматически генерируемый номер |
| Вес голоса | 1 | Зависит от количества помещений пользователя |  | 8 |  |
| **Вариант голосования** |  |  | Идентификатор варианта голосования  Описание |  |  |
| Идентификатор варианта голосования | - | Уникальный идентификатор объекта класса «вариант голосования» |  | 8 | Автоматически генерируемый номер |
| Описание | - | Текстовое описание варианта голосования |  | 4096 | Текстовая строка |
| **Пользователь** |  |  | Идентификатор пользователя  Логин  Пароль  ФИО  e-mail  Телефон  Изображение профиля  Тип пользователя  Признак блокировки |  |  |
| Идентификатор пользователя | - | Уникальный идентификатор объекта класса «пользователь» |  | 8 | Автоматически генерируемый номер |
| Логин | - | Логин пользователя |  | 256 | Текстовая строка |
| Пароль | - | Пароль пользователя |  | 256 | Текстовая строка |
| ФИО | - | ФИО пользователя |  | 256 | Текстовая строка |
| e-mail | - | e-mail пользователя |  | 256 | Текстовая строка |
| Телефон | - | Телефон пользователя |  | 16 | Текстовая строка |
| Изображение профиля | - | Файл изображения |  |  | Графический файл |
| Тип пользователя | false («Собственник») | Собственник или представитель управляющей организации |  | 1 | Логический тип |
| Признак блокировки | false | Показывает, заблокирована УЗ пользователя или нет |  | 1 | Логический тип |
| **Список пользователей** |  |  | Множество пользователей |  |  |
| Множество пользователей | - | Структура, содержащая объекты класса «пользователь» |  |  |  |
| **Список заявок пользователя** |  |  | Пользователь  Множество открытых заявок  Множество закрытых заявок |  |  |
| Пользователь | - | Идентификатор, ссылающийся на пользователя |  | 8 | Автоматически генерируемый номер |
| Множество открытых заявок | - | Структура, содержащая объекты класса «заявка» |  |  |  |
| Множество закрытых заявок |  | Структура, содержащая объекты класса «заявка» |  |  |  |
| **Помещение** |  |  | Идентификатор помещения  Пользователь  Номер  Подъезд  Этаж  Площадь  Тип помещения |  |  |
| Идентификатор помещения | - | Уникальный идентификатор объекта класса «помещение» |  | 8 | Автоматически генерируемый номер |
| Пользователи | - | Структура, содержащая объекты класса «пользователь» |  |  |  |
| Номер | 1 | Номер помещения |  | 8 | Целое число |
| Подъезд | 1 | Номер подъезда |  | 2 | Целое число |
| Этаж | 1 | Номер этажа |  | 4 | Целое число |
| Площадь | 0 | Площадь помещения в кв. м |  | 4 | Целое число |
| Тип помещения | false («Жилое») | Тип помещения |  | 1 | Логический тип |
| **Список помещений** |  |  | Множество помещений |  |  |
| Множество помещений | - | Множество объектов класса «помещение» |  |  |  |

## Архитектура системы

Разработка информационной системы предполагает выбор подходящей архитектуры. Одной из используемых в современных веб-приложениях концепций является MVC – Model-View-Controller. Суть её заключается в разделении бизнес-логики и данных приложения на три слоя[https://brainify.ru/r/article/mvc-dlya-chaynikov/]:

Model (Модель) – предоставляет данные и правила для работы с ними.

View (Представление) – определяет способы представления данных.

Controller (Контроллер) – интерпретирует запросы пользователя (HTTP GET или POST).

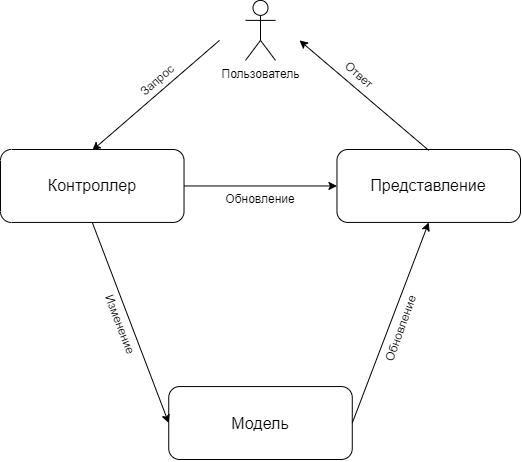


Рис 5. Концепция Модель-Представление-Контроллер

Основным преимуществом данной архитектуры является простота отладки приложения. Например, при возникновении ошибки, связанной с логикой данных, можно сразу отбросить код, относящийся к представлениям и контроллерам, и более оперативно устранить проблему. Кроме этого, отделение бизнес-логики от её визуализации предоставляет более широкие возможности для повторного использования кода, что может пригодиться в случае необходимости представлять данные приложения в различных контекстах – таких, например, как основное представление для непосредственной работы с данными и дополнительное, в виде статистической панели.

Концепция MVC не имеет чёткого определения, где именно располагается бизнес-логика приложения – в контроллере или в модели. Для более чёткого определения архитектуры разрабатываемой системы разобьём её на три уровня – уровень представления, уровень бизнес-логики и уровень хранения данных.

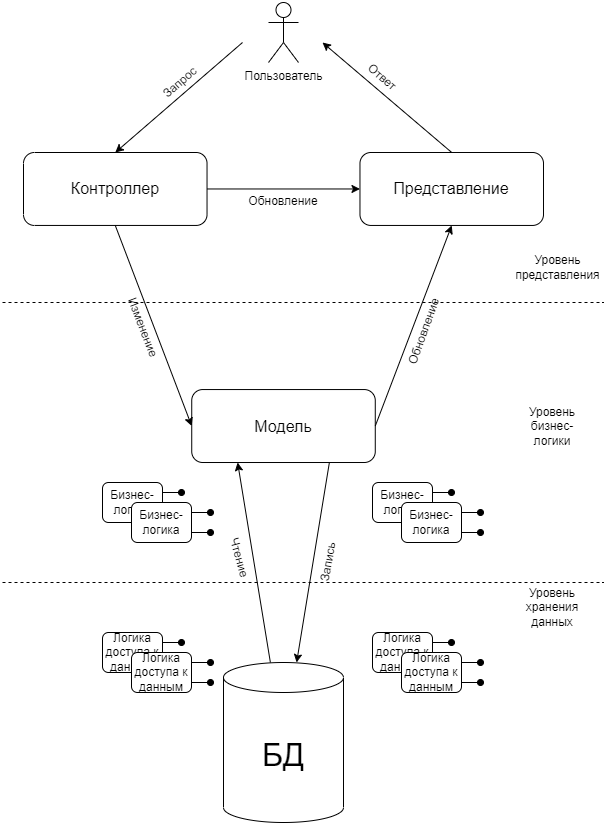


Рис 6. Трёхуровневая архитектура с использованием концепции MVC

Таким образом, контроллер и представление находятся на уровне представления, а бизнес-логика с моделью данных расположены на уровне бизнес-логики. Контроллеры не несут в себе бизнес-логику, и отвечают только за приём и анализ запросов пользователя, и передачу этих запросов другим элементам системы.

Непосредственная реализация трёхуровневой архитектуры в разрабатываемом веб-приложении, с учётом выбранных инструментальных средств, выглядит следующим образом:

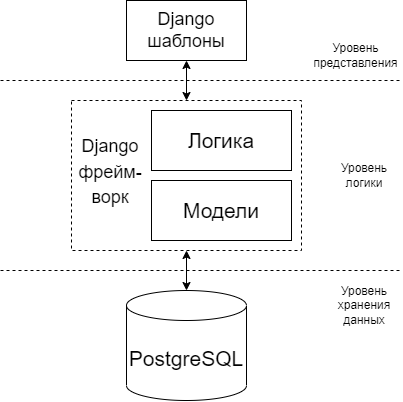


Рис 7. Архитектура веб-приложения

## База данных системы

Атрибуты, перечисленные в словаре данных, используются для проектирования базы данных системы. На рисунке 5 представлена ER-диаграмма, или диаграмма «сущность-связь», которая описывает таблицы базы данных и типы связей между, существующих ними.

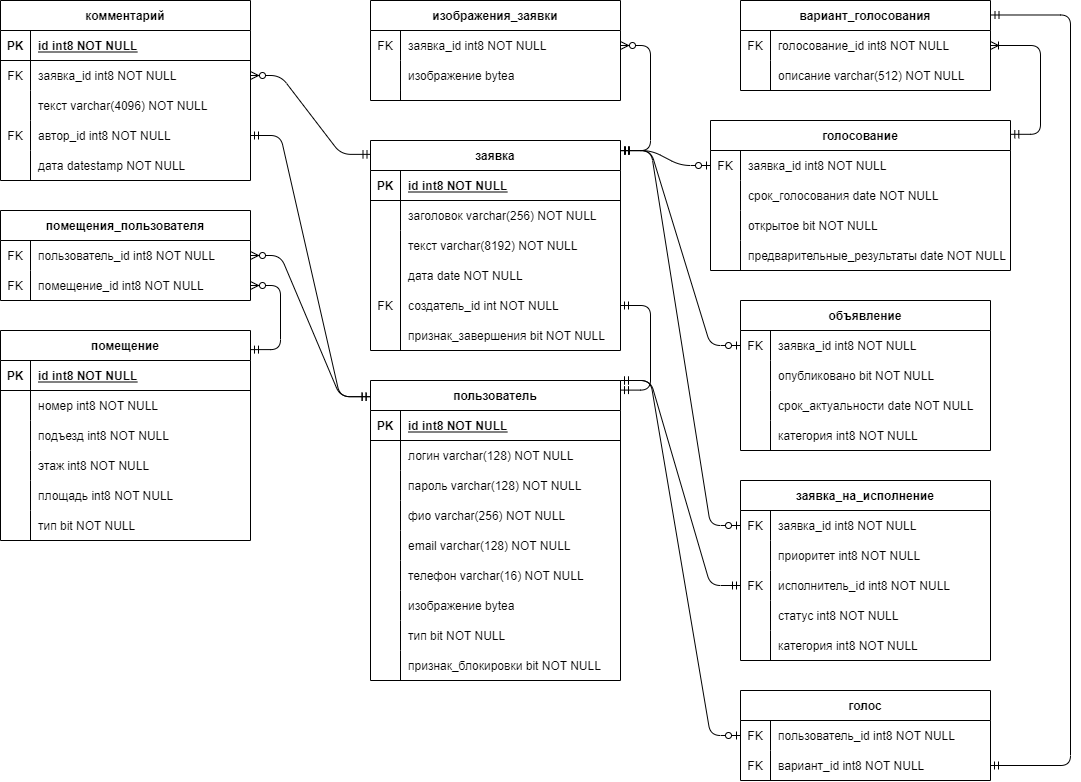


Рис 5. ER-диаграмма системы

Таблицы базы данных содержат атрибуты, указанные в словаре данных.

Таблица 7. Таблица «Заявка»

|  |  |
| --- | --- |
| **Атрибут** | **Значение** |
| ID | Первичный ключ таблицы |
| Заголовок | Заголовок заявки |
| Текст | Текст заявки |
| Дата | Дата создания заявки |
| ID создателя | Создатель заявки. Внешний ключ, ссылка на таблицу «Пользователь» |
| Признак завершения | Логическое поле, признак перехода заявки в статус «Завершена» |

Таблица 8. Таблица «Пользователь»

|  |  |
| --- | --- |
| **Атрибут** | **Значение** |
| ID | Первичный ключ таблицы |
| Логин | Логин пользователя |
| Пароль | Пароль пользователя |
| ФИО | ФИО пользователя |
| Email | Адрес электронной почты пользователя |
| Телефон | Номер телефона пользователя |
| Изображение | Изображение профиля пользователя |
| Тип | Логическое поле. Тип пользователя – «собственник» или «представитель управляющей организации» |
| Признак блокировки | Логическое поле, признак блокировки учётной записи пользователя |

Таблица 9. Таблица «Помещение»

|  |  |
| --- | --- |
| **Атрибут** | **Значение** |
| ID | Первичный ключ таблицы |
| Номер | Номер помещения |
| Подъезд | Номер подъезда, в котором находится помещение |
| Этаж | Номер этажа, на котором находится помещение |
| Площадь | Площадь помещения |
| Тип | Логическое поле, тип помещения – «жилое» или «коммерческое» |

Таблица 10. Таблица «Комментарий»

|  |  |
| --- | --- |
| **Атрибут** | **Значение** |
| ID | Первичный ключ таблицы |
| ID заявки | Заявка, в которой находится комментарий. Внешний ключ, ссылка на таблицу «Заявка» |
| Текст | Текст комментария |
| ID автора | Автор комментария. Внешний ключ, ссылка на таблицу «Пользователь» |
| Дата | Дата комментария |

Таблица 11. Таблица «Заявка на исполнение»

|  |  |
| --- | --- |
| **Атрибут** | **Значение** |
| ID Заявки | Заявка, в рамках которой требуется исполнение. Внешний ключ, ссылка на таблицу «Заявка» |
| Приоритет | Приоритет исполнения заявки |
| ID исполнителя | Текущий исполнитель заявки. Внешний ключ, ссылка на таблицу «Пользователь» |
| Статус | Текущий статус заявки |
| Категория | Категория заявки |

Таблица 12. Таблица «Объявление»

|  |  |
| --- | --- |
| **Атрибут** | **Значение** |
| ID заявки | Заявка, в рамках которой подаётся объявление. Внешний ключ, ссылка на таблицу «Заявка» |
| Опубликовано | Признак публикации объявления |
| Срок актуальности | Срок, по достижении которого объявление теряет актуальность и снимается с публикации |
| Срок актуальности | Категория объявления |

Таблица 13. Таблица «Голосование»

|  |  |
| --- | --- |
| **Атрибут** | **Значение** |
| ID заявки | Заявка, в рамках которой создаётся голосование. Внешний ключ, ссылка на таблицу «Заявка» |
| Срок голосования | Срок, по достижении которого голосование завершается |
| Открытое | Признак того, что голосование является открытым |
| Предварительные результаты | Признак того, что результаты голосования видны до его завершения |

Таблица 14. Таблица «Помещения пользователя»

|  |  |
| --- | --- |
| **Атрибут** | **Значение** |
| ID пользователя | Пользователь, владеющий данным помещением. Внешний ключ, ссылка на таблицу «Пользователь». Часть составного первичного ключа таблицы. |
| ID помещения | Помещение, которым владеет данный пользователь. Внешний ключ, ссылка на таблицу «Помещение». Часть составного первичного ключа таблицы. |

Таблица 15. Таблица «Изображения заявки»

|  |  |
| --- | --- |
| **Атрибут** | **Значение** |
| ID заявки | Заявка, к которой прикреплено данное изображение. Внешний ключ, ссылка на таблицу «Заявка». Часть составного первичного ключа таблицы. |
| ID изображения | Изображение, которое прикреплено к данной заявке. Внешний ключ, ссылка на таблицу «Изображение». Часть составного первичного ключа таблицы. |

Таблица 16. Таблица «Вариант голосования»

|  |  |
| --- | --- |
| **Атрибут** | **Значение** |
| ID голосования | Голосование, к которому относится данный вариант. Внешний ключ, ссылка на таблицу «Голосование». |
| Описание варианта | Текст, суть варианта голосования |

Таблица 17. Таблица «Голос»

|  |  |
| --- | --- |
| **Атрибут** | **Значение** |
| ID пользователя | Пользователь, выбравший данный вариант голосования. Внешний ключ, ссылка на таблицу «Пользователь». Часть составного первичного ключа таблицы. |
| ID варианта голосования | Вариант голосования, выбранный данным пользователем. Внешний ключ, ссылка на таблицу «Вариант голосования». Часть составного первичного ключа таблицы. |

## Проектирование интерфейса системы

Интерфейс системы спроектирован по аналогии с существующими хелпдеск-системами. На рисунке 6 представлен интерфейс окна авторизации.

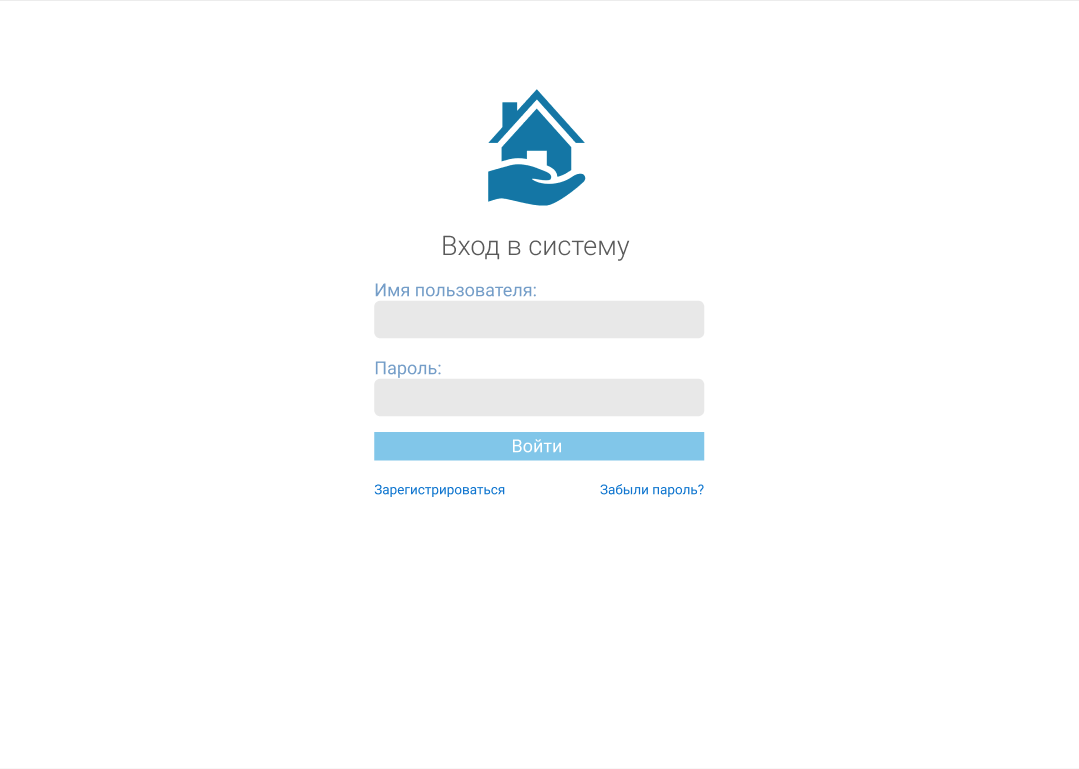


Рис 6. Окно авторизации системы

Заявки отображаются списком. С помощью бокового меню можно выбрать отображение заявок на исполнение, объявлений либо голосований. Так же можно вывести заявки в соответствии с их статусом, например приостановленные заявки или прошедшие голосования. Основное окно, помимо списка заявок, содержит кнопку для создания новой заявки, а также ссылку на профиль пользователя. Заявки в списке содержат имя создателя, заголовок, дату создания, приоритет, тип категорию заявки и текущий статус.

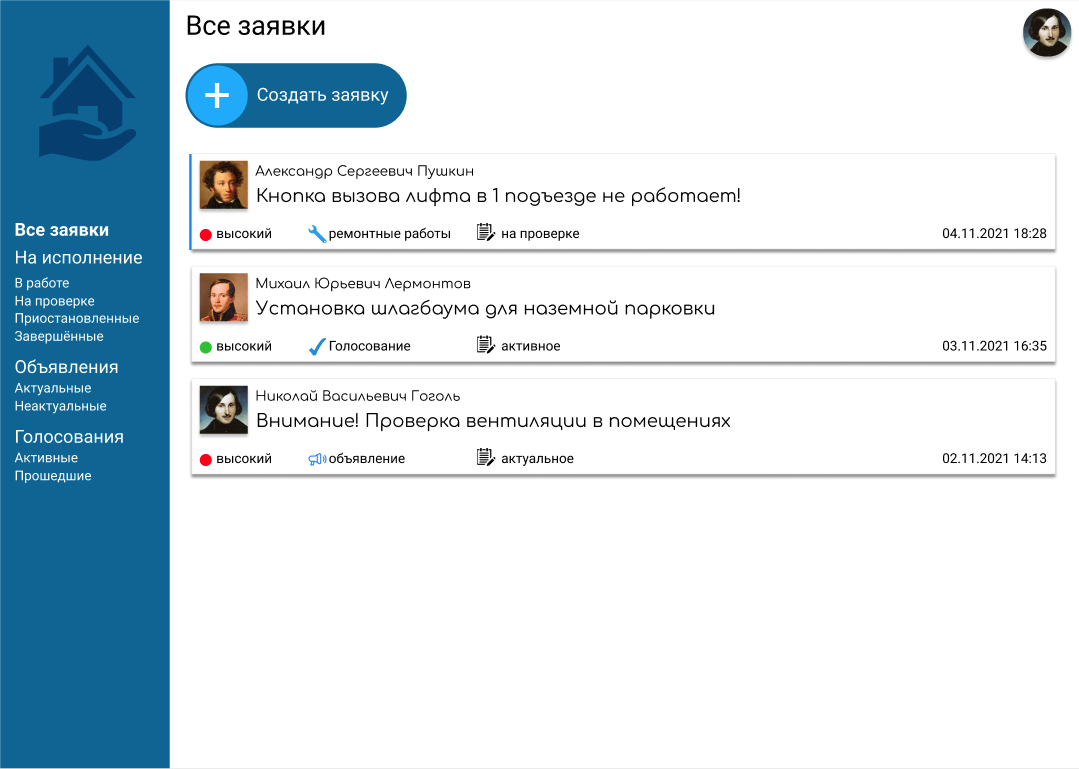


Рис 7. Список заявок пользователя

Форма создания заявки динамически изменяется в соответствии с выбранным типом заявки. Каждому типу – заявкам на исполнение, объявлениям и голосованиям соответствует свой набор полей для заполнения. В заявках на исполнение предоставляется возможность выбрать необходимую категорию, например «лифт», «сантехника», «ремонтные работы» и др. При создании голосования, варианты записываются в одно текстовое поле через точку с запятой. Для голосования и объявления необходимо так же указать дату завершения.

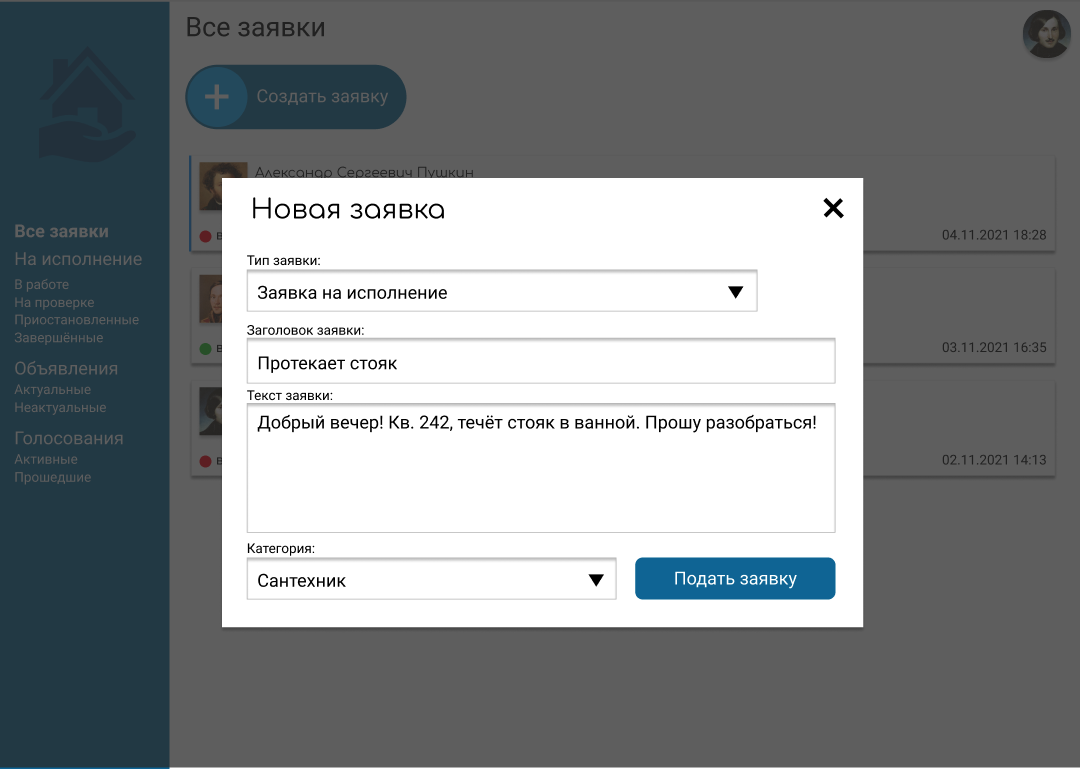


Рис 8. Создание заявки на исполнение

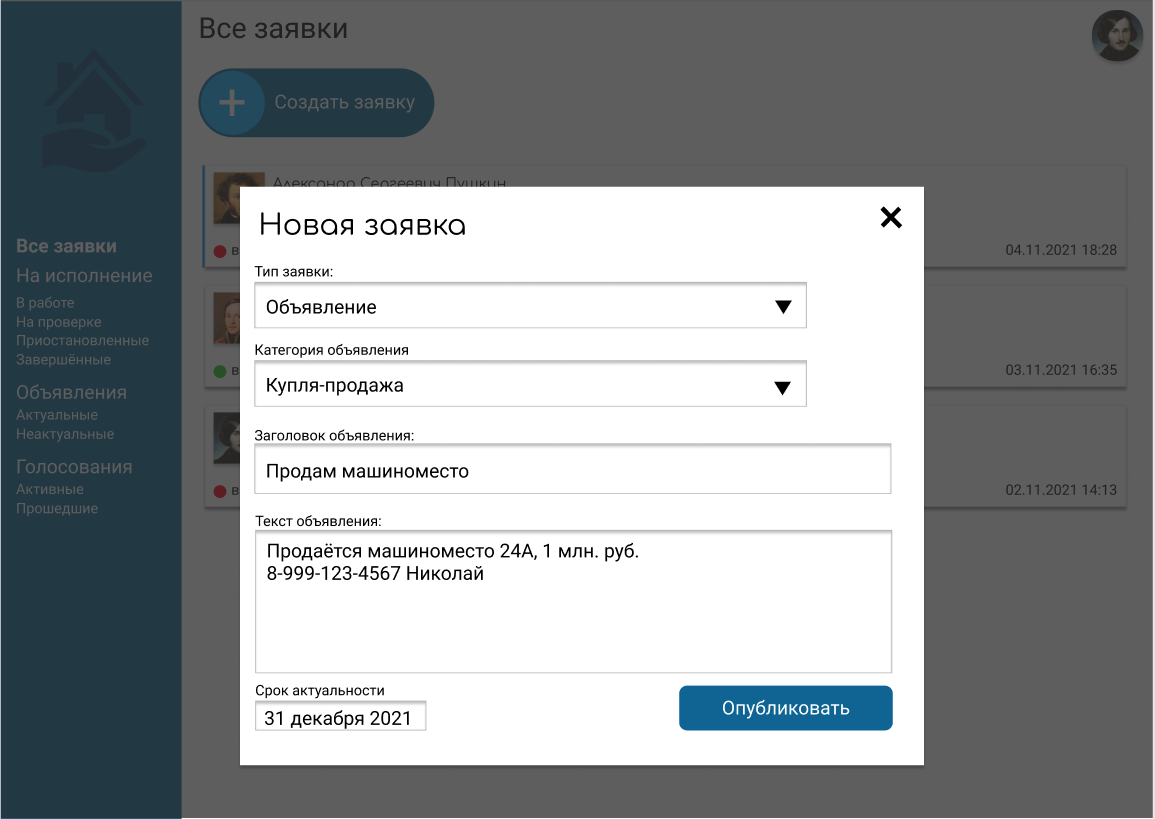


Рис 9. Создание объявления

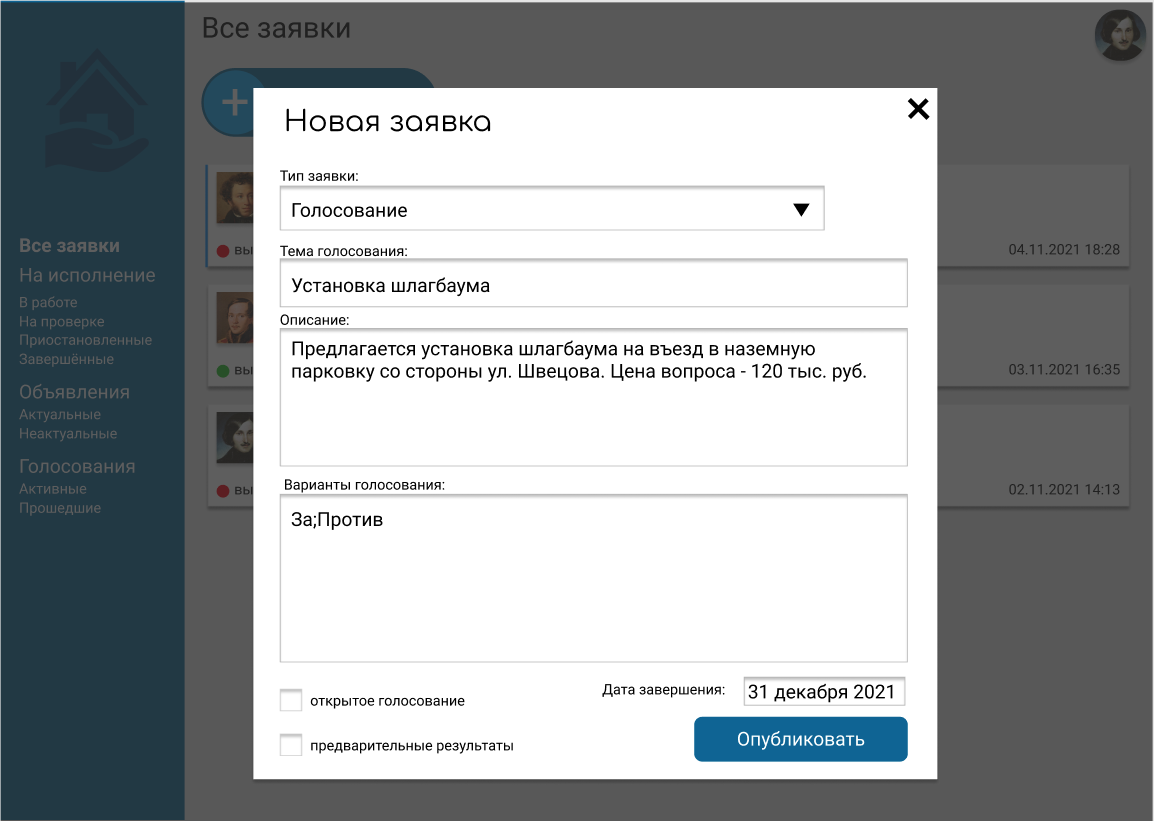


Рис 10. Создание голосования

Аналогичным образом отличаются и уже созданные заявки на исполнение, голосования и объявления. Заявки на исполнение имеют набор элементов интерфейса для смены статуса и комментирования. Объявления имеют только возможность просмотра.

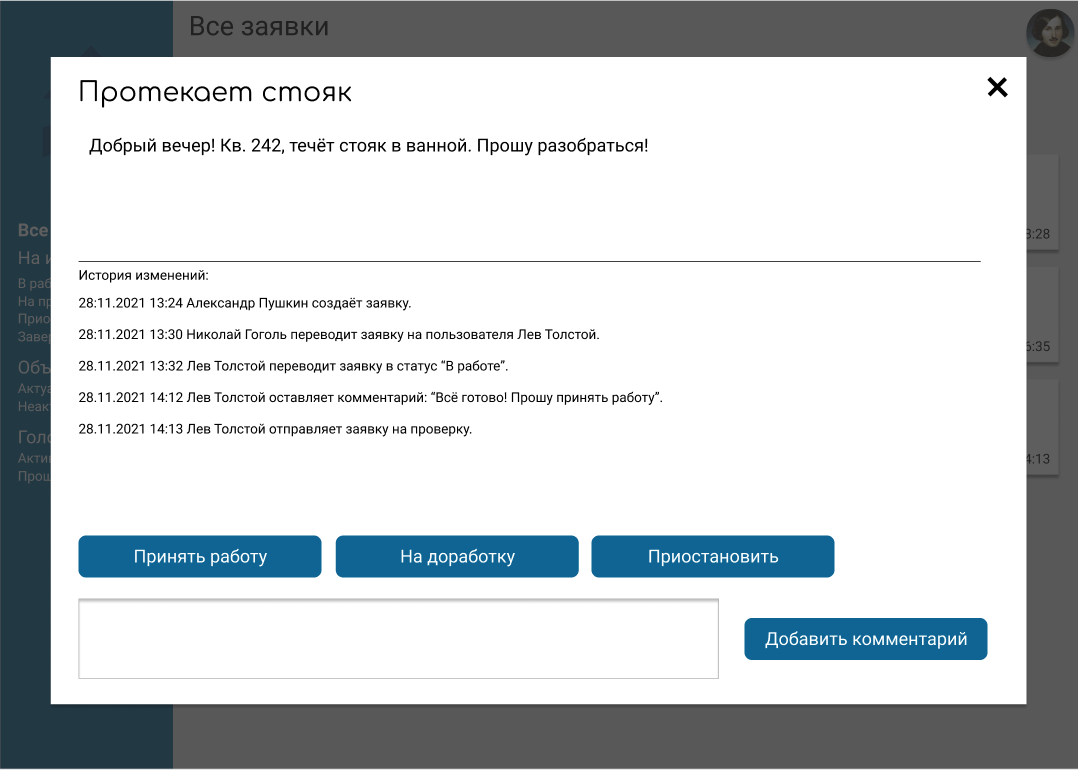


Рис 11. Работа с заявкой на исполнение

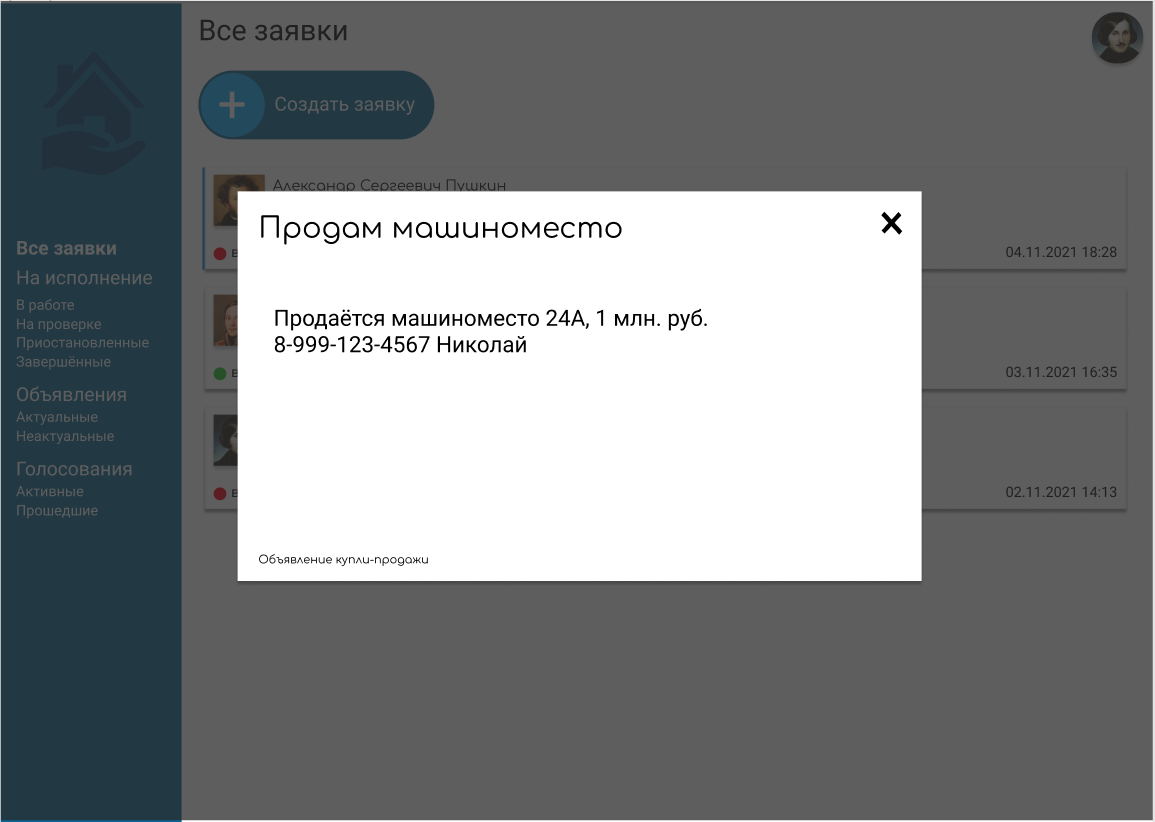


Рис 12. Просмотр объявления

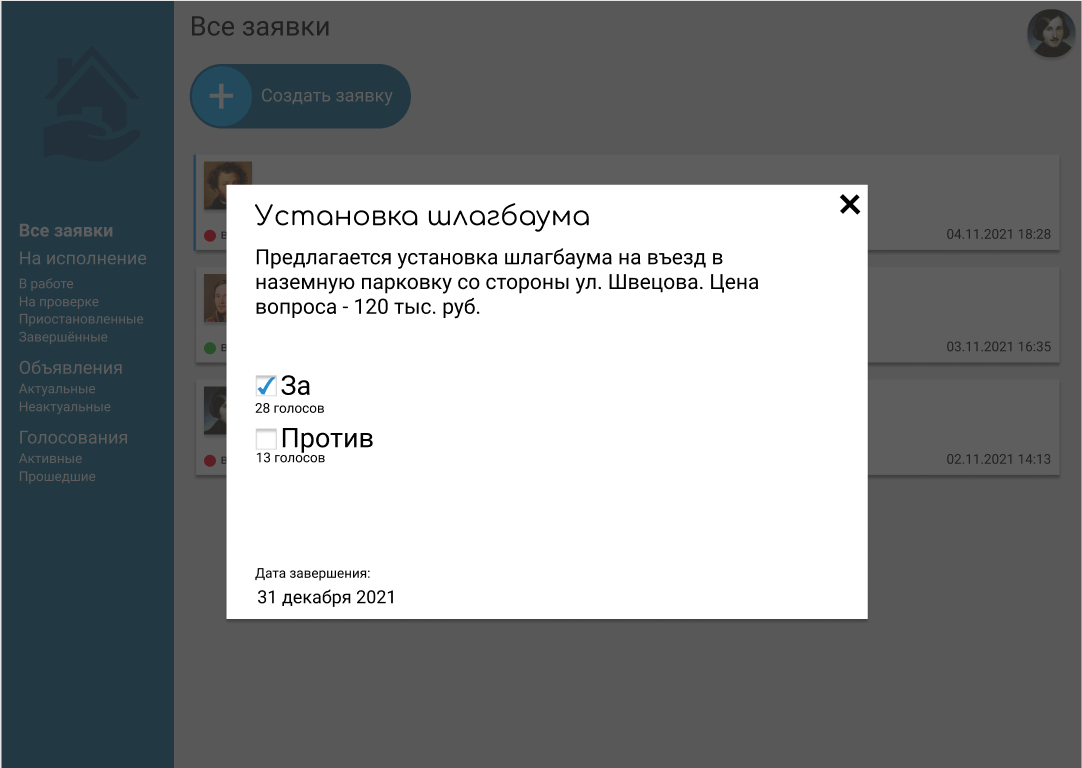


Рис 13. Участие в голосовании

# Разработка системы

## Выбор инструментальных средств разработки

Прежде чем приступить к написанию кода системы, необходимо правильно подойти к выбору программных средств разработки, которые должны отвечать следующим требованиям:

1. Наличие опыта работы с ними, либо относительная простота освоения. Важное требование в условиях ограниченного времени на разработку.
2. Кроссплатформенность и поддержка перспективных систем, регулярные обновления. Система, разработанная с использованием данных средств, должна иметь возможность быть адаптированной для новых версий целевой платформы при минимальных затратах. Использование устаревших средств разработки недопустимо в стремительно развивающемся мире информационных технологий.
3. Бесплатная лицензия.
4. Активное сообщество разработчиков. Должна иметься возможность решать возникающие при работе со средством разработки проблемы, используя ответы на соответствующих ресурсах.

На текущий момент двумя наиболее часто рекомендуемыми для разработки веб-систем фреймворками являются ASP.NET Core и Django.

ASP.NET Core, разрабатываемый компанией Microsoft, – это свободно-распространяемый фреймворк, который применяется для разработки веб-приложений на платформе .NET. Платформа имеет модульную структуру, а в список поддерживаемых операционных систем входят Microsoft Windows, Linux и macOS.

Django – это фреймворк на языке Python, так же применяемый для разработки веб-приложений. Организация Django Software Foundation занимается его поддержкой. Django бесплатен, с открытым исходным кодом и имеет активное и растущее сообщество.

Принимая во внимание такой аспект, как скорость написания веб-приложений, ASP.NET Core не будет являться наиболее оптимальным выбором. При всех его преимуществах, ASP.NET Core требует от разработчика уверенного знания языка C#, а также больших временных затрат на подключение описание и подключение интерфейсов. Django, напротив, позволяет получить готовый результат за относительно небольшой промежуток времени. Такой недостаток Django, как меньшая, по сравнению с основным конкурентом, производительность не будет иметь значения при предполагаемом масштабе использования системы.

## Написание моделей веб-приложения

База данных разрабатываемого приложения реализована на СУБД PostgreSQL. Фреймворк Django работает с БД без прямого участия разработчика, все записи строятся на основе моделей, которые определены в файле models.py. Используя архитектуру БД, описанную ранее посредством ER-диаграммы, были реализованы следующие модели данных:

Модели основной заявки и заявки на исполнение. Атрибуты «Статус», «Приоритет» и «Категория» принимают значение из заранее заданных списков констант. Атрибут «Автор» ссылается на пользователя.

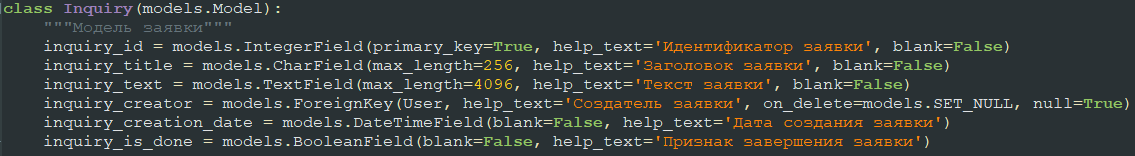


Рис 14. Модель основной заявки

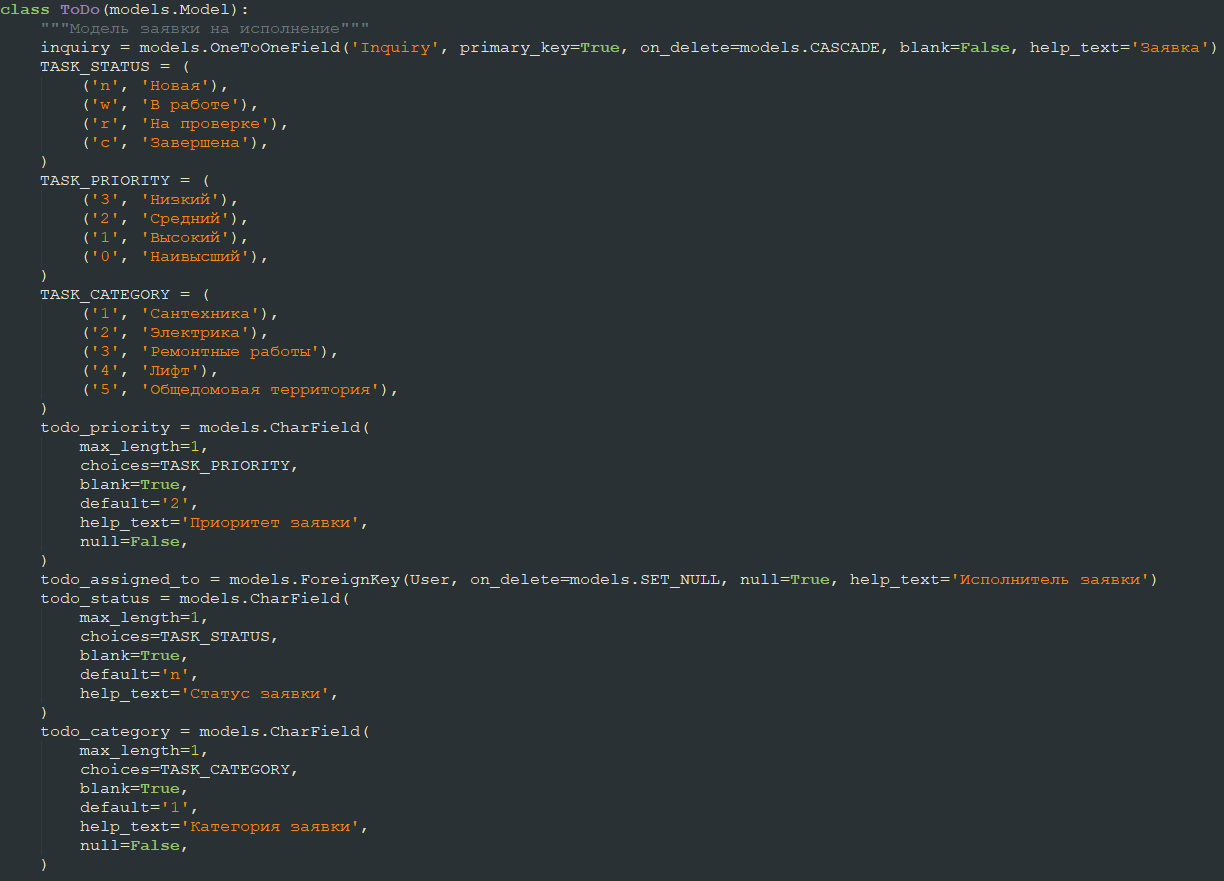


Рис 15. Модель заявки на исполнение

Модель объявления имеет ссылку на голосование, а также предварительно заданные константы категорий объявления. Заголовки категорий являются временными заглушками на текущий момент.

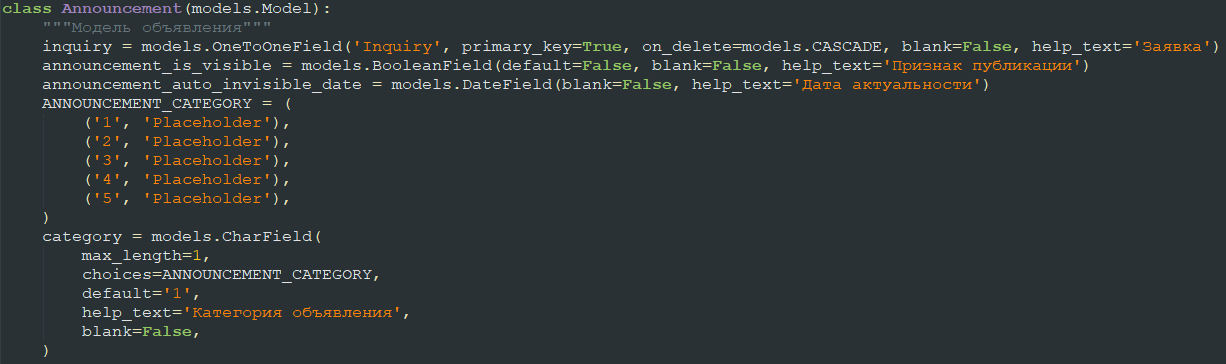


Рис 16. Модель объявления

Модель голосования ссылается на основную заявку и содержит остальные атрибуты, заданные исходя из описания соответствующего класса. Модель голоса ссылается на соответствующий вариант голосования, а модель варианта голосования, в свою очередь имеет ссылку на голосование.

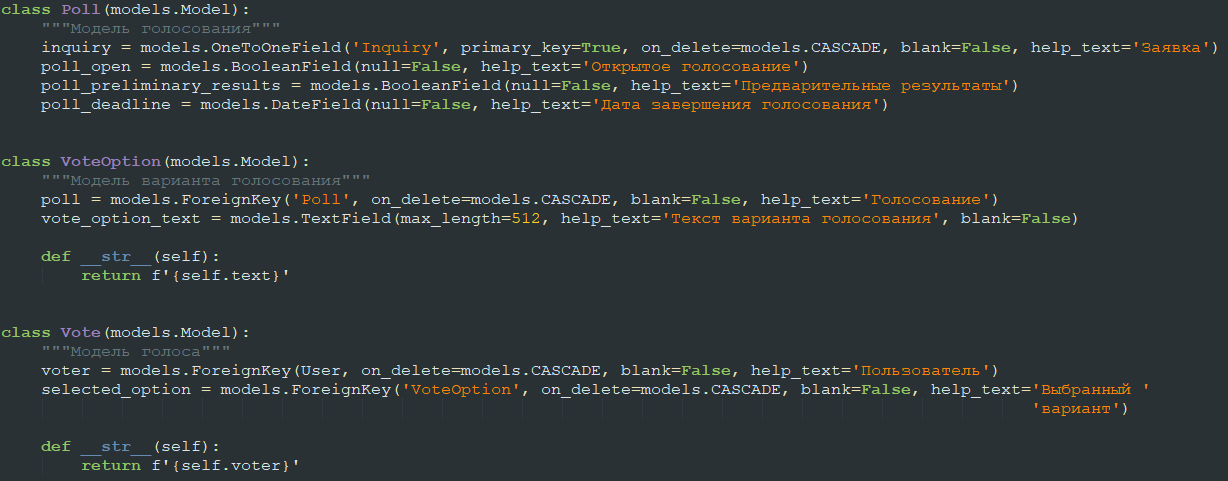


Рис 17. Модели голосования, варианта голосования и голоса

Модель помещения подразумевает два значения атрибута «тип» – «жилое» и «коммерческое». Эти значения выбираются из заданных констант. Так же, в модели указаны все необходимые атрибуты – адрес дома, номер подъезда, этажа и пр.

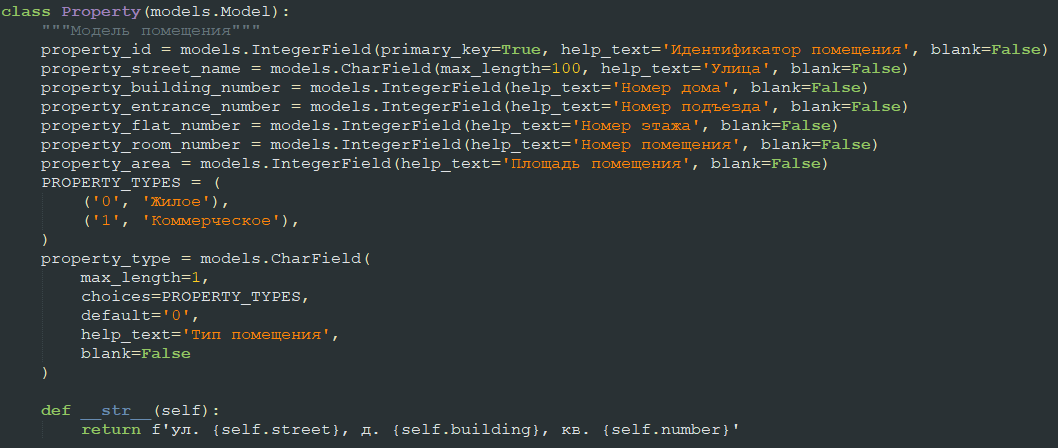


Рис 16. Модель помещения

Модель комментария через связь многие-к-одному ссылаются на заявку. Таким образом, в одной заявке пользователи могут оставлять многочисленные комментарии. Ссылка на пользователя, оставившего комментарий, так же описана в модели.

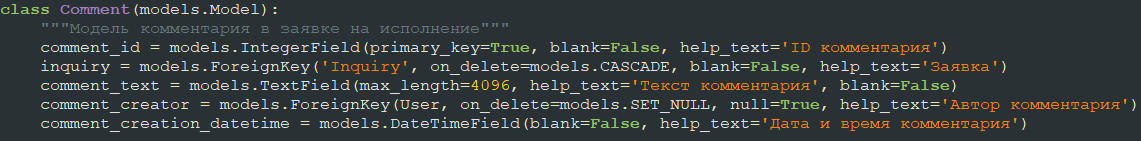


Рис 17. Модель комментария

В Django применяется встроенная модель пользователя – User. Данная модель не имеет полный набор всех необходимых атрибутов, в частности отсутствуют атрибуты для телефона или адреса электронной почты пользователя. Для реализации хранения этих данных в базе была создана модель профиля пользователя, которая ссылается на модель пользователя связью один-к-одному.

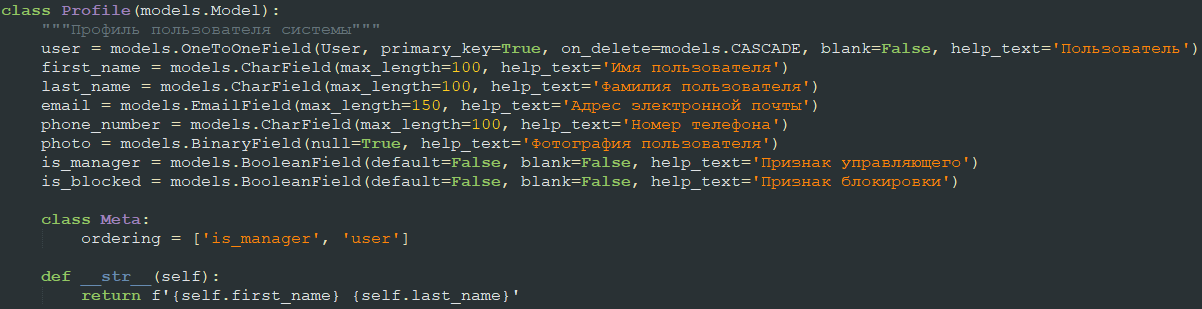


Рис 19. Модель профиля пользователя

# Заключение

В результате проделанной работы были выполнены следующие этапы разработки системы коммуникации для управления многоквартирным домом: