

## СОДЕРЖАНИЕ

Введение.....	7
1 Концептуальные основы анализа рынка покупки недвижимости.....	8
1.1 Основные элементы процесса купли-продажи квартиры.....	8
1.2 Обзор рынка недвижимости.....	11
1.3 Особенности современного состояния рынка недвижимости Республики Беларусь .....	14
1.4 Анализ отдельных сегментов рынка недвижимости Республики Беларусь .....	16
2 Анализ бизнес-процессов, связанных с покупкой квартиры.....	20
2.1 Описание предприятия ОАО «Лидагропромаш» .....	20
2.2 Анализ состояния изучаемого процесса с целью выявления недостатков и обоснования целесообразности автоматизации процесса .....	21
2.3 Описание процесса покупки квартиры as-is .....	23
2.4 Описание процесса покупки квартиры as to be.....	31
2.5 Обзор существующих систем .....	33
2.6 Основные подходы к поддержке принятия решений о приобретении квартиры.....	37
3 Программная реализация системы поддержки принятия решений о приобретении квартиры.....	41
3.1 Постановка задачи .....	41
3.2 Обоснование выбора компонентов и технологий .....	41
3.3 Модели представления системы и их описание .....	43
3.4 Информационная модель системы и её описание .....	49
3.5 Основные алгоритмы программы .....	52
3.6 Руководство пользователя.....	53
4 Экономическое обоснование разработки автоматизированной системы поддержки принятия решений о приобретении квартиры .....	63
4.1 Характеристика программного продукта .....	63
4.2 Расчёт затрат на разработку программного продукта.....	63
4.3 Оценка экономического эффекта от реализации программного продукта .....	68
4.4 Расчёт показателей эффективности инвестиций в разработку ПО.....	69
Заключение .....	72
Список использованных источников .....	73
Приложение А (обязательное) Отчёт о проверке в системе «Антиплагиат»..	75

Приложение Б (обязательное) Основные алгоритмы работы программы .....	76
Приложение В (обязательное) Листинг скрипта генерации базы данных .....	78
Приложение Г (обязательное) Листинг кода .....	81
Ведомость документов.....	87

## ВВЕДЕНИЕ

В современном быстро развивающемся мире, где инновации в технологиях играют ключевую роль, сфера жилья также стремится к автоматизации и оптимизации процессов. Принятие решения о приобретении жилья – важный этап в жизни каждого человека, требующий внимательного анализа и обдуманных шагов. В этом контексте, создание автоматизированной системы поддержки принятия решений о покупке квартиры становится актуальным и перспективным направлением.

Одной из ключевых задач такой системы является анализ и структурирование больших объемов данных, включая информацию о рыночной стоимости, районе, расположении объекта, транспортной инфраструктуре и других факторах, влияющих на принятие решений.

Этот проект не только призван облегчить процесс выбора жилья, но и оптимизировать его, предоставляя пользователям инновационные инструменты и данные для более информированных и обоснованных решений.

Целью данного дипломного проекта является уменьшения затрачиваемого времени на поиск и анализ вариантов жилья за счёт оптимизирование процесса выбора квартиры.

Для выполнения поставленной цели необходимо выполнить следующие задачи:

- исследовать процесс поиска и покупки недвижимости;
- проанализировать бизнес-процессы, связанные с покупкой квартиры;
- разработать автоматизированную систему поддержки принятия решений о приобретении квартиры;
- провести экономическое обоснование эффективности разработки и реализации программного продукта.

Объектом исследования является процесс приобретения. Предмет исследования – методы и технологии для решения вопросов о приобретении квартиры, метод экспертных оценок. Дипломный проект выполнен самостоятельно, проверен в системе «Антиплагиат». Процент оригинальности составляет 98,18%. Цитирования обозначены ссылками на публикации, указанными в «Списке использованных источников». Скриншот приведен в приложении А (рисунок А.1).

# **1 КОНЦЕПТУАЛЬНЫЕ ОСНОВЫ АНАЛИЗА РЫНКА ПОКУПКИ НЕДВИЖИМОСТИ**

## **1.1 Основные элементы процесса купли-продажи квартиры**

Недвижимость составляет большую часть национального богатства любой страны, большую часть муниципальной и государственной собственности, а в масштабах отдельных предприятий, за редким исключением, существенную часть их имущества. Кроме того, организация любой производственной либо коммерческой деятельности требует наличия соответствующего помещения, что делает возможным отнести недвижимость к разряду факторов производства. Вышеизложенное позволяет проиллюстрировать тот факт, что недвижимость выступает в качестве важнейшего объекта управления как для государственных и муниципальных органов власти, так и для отдельных предприятий. В то же время сложившаяся система управления недвижимостью является неэффективной, что обуславливает необходимость ее совершенствования. Рынок недвижимости является достаточно специфическим видом рынка. Ему присущ ряд особенностей, нехарактерных для других рынков. В частности, это касается законодательного регулирования рынка недвижимости, особенностей определения рыночной стоимости недвижимости, вопросов финансирования рынка недвижимости, особенностей организации деятельности его профессиональных участников. К недвижимым вещам относятся земельные участки, участки недр, поверхностные водные объекты и все, что прочно связано с землей, то есть объекты, перемещение которых без несоразмерного ущерба их назначению невозможно, в том числе леса, многолетние насаждения, капитальные строения (здания, сооружения), незавершенные законсервированные капитальные строения, изолированные помещения, машино-места. К недвижимым вещам также приравниваются предприятие в целом как имущественный комплекс, подлежащие государственной регистрации воздушные и морские суда, суда внутреннего плавания, суда плавания «рекаморе», космические объекты. Законодательными актами к недвижимым вещам может быть отнесено и иное имущество. Недвижимое имущество как товар – это объект сделок, удовлетворяющий различные реальные или потенциальные потребности и имеющий определенные качественные и количественные характеристики. Однако недвижимость – это особый товар, который обладает целым рядом характерных только для него свойств. Любой объект недвижимости в реальной действительности

существует в единстве физических, экономических, социальных и правовых свойств, каждое из которых может в соответствующем случае выступать в качестве основного в зависимости от жизненных ситуаций, целей и стадий анализа. В экономическом аспекте недвижимость представляет собой источник дохода, средство обеспечения финансовых инструментов, а также как объект, требующий определенного потока инвестиций, направленных на его сохранение и развитие. За счет налогообложения владельцев (пользователей) недвижимости формируются государственные бюджеты и реализуются социальные программы.

Но тем не менее, не смотря на всю сложность сферы недвижимости и её многообразие, в процессе покупки квартиры можно выделить следующие этапы [1]:

1 Определиться с основными критериями выбора жилого помещения. На этом этапе потенциальный покупатель определяется с параметрами квартиры, которые ему наиболее и наименее важны, основываясь на своих предпочтениях и потребностях. Такими параметрами могут быть: близость к метро, близость к школе, район, в котором находится квартира, новостройка ли это или вторичка. Также на этом этапе определяются финансовые возможности, рассматривается доступная сумма, возможности кредитования и иные финансовые аспекты.

2 Найти, осмотреть и выбрать для себя подходящий вариант. На этом этапе задействуются различные способы поиска квартиры, начиная от рекомендаций знакомых, заканчивая различными сайтами и агентствами по недвижимости. Используя полученную информация, проведя все доступные квартиры через фильтр предпочтений и пройдя предварительный осмотр квартиры, покупатель, возможно проведя сравнительный анализ нескольких понравившихся вариантов, выбирает вариант квартиры, который ему понравился.

3 Обратиться вместе с продавцом в агентство по государственной регистрации недвижимого имущества или к нотариусу за консультацией. На этом этапе дело доходит до правовой части, в которой происходит проверка юридической чистоты объекта недвижимости, и подготавливается необходимый пакет документов для сделки. Для этого этапа понадобятся услуги соответствующих людей и организации, таких как республиканская организация по государственной регистрации и консультация нотариуса. В момент удостоверения договора купли-продажи недвижимости нотариус или регистратор несет ответственность за законность сделки. Они проверяют все документы, необходимые для проведения этой процедуры. В случае, если

каких-то данных или документов нотариусу или регистратору не будет достаточно, то сделка не будет произведена до момента их получения, уточнения. Поэтому и покупателю, и продавцу важно подготовиться к заключению договора купли-продажи, собрать необходимые документы.

4 Составить предварительный акт приема-передачи. В акте приема-передачи будет указано, что продавец обязуется передать товар в собственность покупателю, а тот в свою очередь принимает этот товар и платит за него определенную денежную сумму, а также что должно остаться в квартире в момент ее передачи новому владельцу, здесь же можно указать показания приборов учета таких как газ, вода, электричество [2].

5 Записаться заранее на заключение сделки. Заключается и удостоверяется договор купли-продажи квартиры у нотариуса либо у регистратора в агентстве по госрегистрации.

6 Произвести расчет по сделке любым законным способом. Перевод может быть осуществлён как наличными, так и осуществить перевод на расчетный счет, открытый на имя продавца. Расчет по сделке происходит белорусских рублях и подтверждается путем подписания расписки о получении денег. Расчёт может быть произведён до или после подписания договора.

7 Получить квартиру покупателю. Момент фактического перехода собственности новому владельцу оформляется документально. Стоит указать в договоре купли-продажи, когда именно жилье будет передано покупателю, или же отразить тот факт, что оно уже передано. Также есть передаточный акт, который подписывается обеими сторонами, когда одна передает, а другая – получает помещение, именуемый актом приёма-передачи.

8 Зарегистрировать договор купли-продажи и переход права собственности в агентстве по госрегистрации.

9 После этого новый собственник жилья может им распоряжаться. Например, регистрироваться по месту жительства и регистрировать своих членов семьи и совершать иные сделки, предусмотренные законодательством.

Из вышеперечисленных действий и состоит процесс купли-продажи недвижимости в большей или в меньшей мере. Каждый из этих элементов играет ключевую роль в успешном завершении процесса купли-продажи квартиры, обеспечивая защиту интересов сторон и соблюдение законодательства.

Рассмотрим более детально второй шаг покупки квартиры. Когда потенциальный покупатель определился, что ему важно он идёт смотреть квартиры и самый популярный способ это через сайты по продаже квартир. И

он сталкивается с проблемой, что невозможно учесть все его пожелания, либо же сайт не предоставляет удобного интерфейса для демонстрации характеристик и предоставления самых подходящих под конкретный запрос квартир. В общем и целом, эти действия по поиску и анализу квартиры занимают довольно большое количество времени и тем самым могут уменьшить желание покупки квартиры или же вообще оттолкнуть от принятия данного решения. Также, к примеру, квартиры могут быть одинаковы по некоторым параметрам, но из-за различия планировки, ощущение метража могут быть разные. Поэтому нужно некое решение, которое удовлетворяло бы эти запросы и автоматизировало данные процессы. Закономерно понятно, что ощущения и привлекательность квартиры субъективны, но если квартиры нравится большинству, то высока вероятность, что и отдельно взятому человеку тоже понравится.

## **1.2 Обзор рынка недвижимости**

Основой любой социально-экономической системы является собственность. Переход общества на рыночные основы вызывает ее трансформацию: собственность из государственной преобразуется в многоаспектный комплекс, состоящий из муниципальной, частной и других видов собственности. Таким образом, меняется механизм управления собственностью, и, как следствие, зарождаются и формируются различные рынки. В условиях становления рыночной экономики рынок приобретает все большее значение. В современном понимании рынок представляет собой конкретную форму проявления товарного обмена и обращения объектов этого рынка, совокупность экономических отношений и связей между покупателями и продавцами, а также торговыми посредниками по поводу движения товаров и денег.

Не так много в экономике отраслей, от нормального функционирования которых зависела бы жизнь всех граждан страны. Строительная отрасль – одна из них, к тому же она напрямую связана с понятием рынка недвижимости. Объекты недвижимого имущества занимают особое место в любой системе общественных, экономических и социальных отношений и при любом общественном устройстве, т. к. прямо или косвенно с ними связаны и хозяйственная деятельность любого предприятия, и интересы людей во всех сферах жизнедеятельности.

Под рынком недвижимости понимается определенный набор механизмов, посредством которых передаются права на собственность и

связанные с ней интересы, устанавливаются цены и распределяется пространство между различными конкурирующими вариантами землепользования. Рынок недвижимости представляет собой сферу вложения капитала в объекты недвижимости и системы экономических отношений, возникающих при операциях с недвижимостью. Эти отношения появляются при купле - продаже недвижимости, ипотеке, сдаче объектов недвижимости в трастовое управление, аренду, наем и т. д. Таким образом, на рынке недвижимости должны действовать продавцы и покупатели, а также отдельные субъекты предпринимательства, оказывающие им содействие в обращении объектов недвижимости [3].

**История становления и развития рынка недвижимости Республики Беларусь.** Рынок недвижимости Республики Беларусь имеет сравнительно небольшой период развития, однако на сегодняшний день он является одним из самых динамично развивающихся. Экономические процессы, под влиянием которых рынок недвижимости формировался и развивался, накладывали особый отпечаток на взаимоотношения его участников, а также на содержание и характеристики предпринимательской деятельности в сфере недвижимости. Рынок недвижимости развивается по своим законам и по-своему реагирует на процессы, происходящие в обществе и экономике. Он более неповоротлив, чем другие товарные рынки, и реагирует на внешние изменения с некоторым опозданием хотя бы потому, что объекты всех видов недвижимости – дорогостоящий товар, а процесс их приобретения или продажи весьма трудоемок и длителен, а также высоки транзакционные издержки.

Становление белорусского рынка недвижимости – очень сложный и противоречивый процесс, который протекал в сложный период зарождения и формирования рыночных отношений в экономике страны [4].

Первый этап – стихийное зарождение рынка в период 1989-1992 годов. Этот период характеризовался отсутствием законодательной базы для осуществления сделок с недвижимостью, отсутствием профессиональных участников рынка недвижимости, высоким уровнем риска при обороте недвижимого имущества и прав на него, сверхвысокими прибылями, вызывающими спекулятивные интересы недобросовестных субъектов экономической деятельности.

Второй этап – формирование жилищного законодательства и бесплатная приватизация существующего жилищного фонда в 1992-2000 годах. Недвижимость городов и других населенных пунктов, промышленных предприятий и объектов официальной инфраструктуры в результате вовлечения ее в имущественный оборот приобрела главное рыночное



свойство – свойство ликвидности. Этот период характеризуется появлением и развитием профессиональных участников и соответствующей инфраструктуры рынка, формированием отдельных секторов рынка недвижимости: жилого фонда и рынка земельных участков. Более интенсивное развитие получил рынок жилой недвижимости. В качестве основных мотивов приобретения недвижимости выступали потребительские (улучшение жилищных условий лицами, создавшими необходимые накопления в других секторах экономики) и спекулятивные (перепродажа приобретаемой недвижимости в условиях быстрого роста цен давала возможность получить за полгода 50% прибыли, а за год – даже 100%) – мотивы, которые доминировали на белорусском рынке недвижимости в 90-е годы.

Третий – современный этап, на котором формируются механизмы управления отношениями между профессиональными участниками рынка недвижимости и прослеживаются границы самостоятельных сегментов рынка: первичного и вторичного; жилого, коммерческого и земли.

Развитие отечественного рынка недвижимости приходится на начало девяностых годов двадцатого века. Правовым основанием его становления явилось провозглашение Конституцией Республики Беларусь существования наряду с государственной собственностью и частной формы собственности. Этот период характеризуется реформированием земельных отношений, приватизацией государственных предприятий и передачей в собственность граждан занимаемых ими жилых помещений. В это время был принят Закон «О приватизации жилищного фонда в Республике Беларусь», который послужил началом зарождения современного рынка недвижимости в Республике Беларусь. Вместе с этим появились первые легальные сделки по продаже квартир. Этому процессу предшествовал принятый в стране Закон «О выкупе квартир в частную собственность». К тому же процесс приватизации государственной собственности, который осуществлялся с 1992 года, способствовал развитию рынка жилья, формированию рынка нежилой недвижимости и становлению института рыночной оценки недвижимого имущества. Окончательно относительная стабилизация на рынке недвижимости оформилась после снятия в 1995 году ограничения, связанного с невозможностью продажи государственных квартир в течение трех лет.

Начальный период зарождения и формирования отечественного рынка недвижимости характеризуется отсутствием необходимых для его нормального функционирования законодательных норм и процедур, обеспечивающих безопасное совершение сделок, защиту прав собственности.

В какой-то период времени недостаток нормативно-правовых актов тормозил развитие и становление рынка недвижимости в Беларуси.

### **1.3 Особенности современного состояния рынка недвижимости Республики Беларусь**

С течением времени ситуация постоянно менялась, и с тех пор, как рынок недвижимости в нашей стране оформился в самостоятельный сектор экономики произошло много качественных изменений. Учитывая значение этого рынка, государство начало принимать меры, направленные на привлечение на него специалистов, обладающих профессиональными знаниями и умениями, гарантирующих законность совершаемых сделок; активно развивалась законодательная база, регулирующая процессы взаимодействия различных участников рынка. Таким образом, постепенно развивалась инфраструктура функционирования этого рынка, и сформировались профессиональные участники этого сектора экономики. Можно заметить, что и отношение белорусского общества к специалистам, занимающимся деятельностью на рынке недвижимости, сегодня тоже постепенно меняется в лучшую сторону.

Таким образом, можно сказать, что рынок недвижимости в Республике Беларусь активно развивается, профессиональные участники становятся все более значимыми игроками на этом рынке. Высокая социальная роль рынка недвижимости в экономике любой страны требует особого внимания как со стороны государства, предпринимателей, работающих на этом рынке, так и со стороны ученых и исследователей, занимающихся изучением протекающих на нем процессов.

За последние годы рынок недвижимости Республики Беларусь повысил свою активность с помощью действия предприятий, специализирующихся на операциях с недвижимостью, финансовых организаций и агентств недвижимости. В период перехода страны к рыночной экономике многие производственные предприятия, объединения и другие организации стали независимыми. Нормальное их функционирование на рынке недвижимости в подобных условиях практически невозможно без хорошо организованной инфраструктуры предпринимательской деятельности. С развитием рынка недвижимости начинает формироваться более развернутая структура предпринимательской деятельности, появляются новые виды услуг и соответственно специалистов.

В настоящее время в Республике Беларусь на рынке недвижимости, с точки зрения формирования цен, следует выделять две его составляющие: первичный и вторичный рынок. На первичном рынке недвижимость как товар выступает впервые. Основными продавцами недвижимости в таком случае выступают государство в лице своих республиканских, региональных и местных органов власти, а также строительные компании – поставщики жилой и нежилой недвижимости.

На вторичном рынке недвижимость выступает как товар, ранее бывший в употреблении и принадлежащий определенному собственнику – физическому или юридическому лицу. Подобное деление рынка имеет место и на рынке потребительских товаров, рынке ценных бумаг и так далее. Но там товары свободно перемещаются в экономическом пространстве, в то время как предложение на рынке недвижимости всегда привязано к определенной территории (административно-территориальному делению), в рамках города – определенному району или даже микрорайону. Учитывая тот факт, что потребности населения в жилье велики, а предприниматели в производственной недвижимости далеки от удовлетворения данных потребностей, дальнейшее развитие рынка недвижимости связано с новым строительством, следовательно, с более быстрым развитием первичного рынка недвижимости.

Первичный и вторичный рынки в определенной степени противостоят друг другу, но в то же время как бы взаимно дополняют друг друга. Например, если по каким-либо причинам, к примеру, мировой финансовый кризис и как результат спад деловой активности, в регионе увеличивается предложение недвижимости на вторичном рынке, то автоматически падают спрос и цены на первичном. Такая зависимость обусловлена тем, что, выполняя общую функцию по торговле и обращению жилой недвижимости, они используют специфические методы отбора и реализации.

Первичный рынок, как правило, охватывает лишь новую недвижимость, вторичный рынок – старую. В Беларуси преобладает вторичный рынок, хотя в последние годы после интенсивного строительства коммерческой, жилой площади, первичный рынок набрал серьезные обороты.

По структуре рынок недвижимости делится на рынок жилья, рынок коммерческой недвижимости, рынок земли. Более того, рынок недвижимости всегда представляет интерес как для простых людей, так и с точки зрения коммерческой недвижимости – для покупки, аренды производственных площадей, инвестиций и так далее.

В силу специфических особенностей формирования и развития белорусского рынка недвижимости сегменты рынка недвижимости в стране развиты неравномерно: рынок земли практически не развит; рынок коммерческой недвижимости развит слабо; наиболее крупным и наиболее развитым сегментом является сектор жилой недвижимости. Однако следует заметить, что в последнее время государственными органами, профессиональными участниками и некоммерческими профессиональными объединениями рынка недвижимости разрабатывается согласованная стратегия развития цивилизованного рынка недвижимости в стране.

#### **1.4 Анализ отдельных сегментов рынка недвижимости Республики Беларусь**

В соответствии с законодательством жилищный фонд включает в себя все жилые помещения Республики Беларусь независимо от форм собственности. Этот сегмент рынка недвижимости является наиболее развитым в нашей стране, так как именно с вовлечения на рынок жилых объектов стал формироваться собственно рынок недвижимости в независимой Беларуси. Ведь потребность в жилье одна из основных потребностей, которую человек в процессе жизнедеятельности пытается удовлетворить.

Остановимся подробнее на анализе ситуации в городе Минске, как наиболее развитом отечественном сегменте рынка недвижимости Республики Беларусь. Количество желающих приобрести жилье в столице постоянно увеличивается. По данным Национального кадастрового агентства количество сделок купли-продажи квартиры за первый квартал перевалило за четыре тысячи (рисунок 1.1) [5]. Это максимум для начала года за всю историю наблюдений. Март в отдельности также стал рекордным – в Минске продано 1,5 тыс. квартир.

Всего за три месяца в Минске было приобретено 287 тысяч квадратных метров жилья на 206 млн долларов. Эта сумма на 20% превышает показатель аналогичного периода прошлого года.

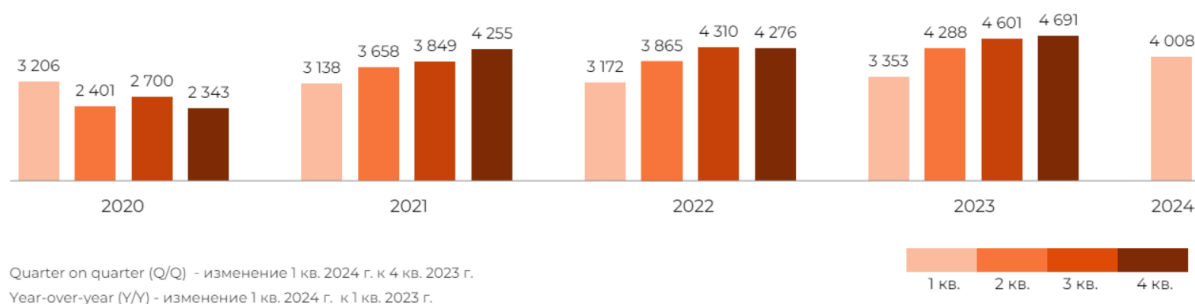


Рисунок 1.1 – Количество сделок с квартирами в Минске

Несмотря на такой высокий спрос предложение квартир в базе Realt по сравнению с предыдущим месяцем выросло на 12%. Однако, это не говорит о том, что выбор квартир в Минске растет. Такой рост связан в первую очередь с большим количеством новых объявлений о продаже квартир в отдельных новостройках. Если из расчетов исключить первичный рынок, то выбор на вторичке, напротив, снижается уже несколько месяцев подряд [6].

На вторичном рынке продолжился плавный рост цен, который начался со 2 квартала 2022 года – динамику иллюстрирует график. Стоимость квадратного метра в однокомнатных квартирах на вторичке превышает 1,5 тыс. долларов. За год она поднялась на 9% (рисунок 1.2).

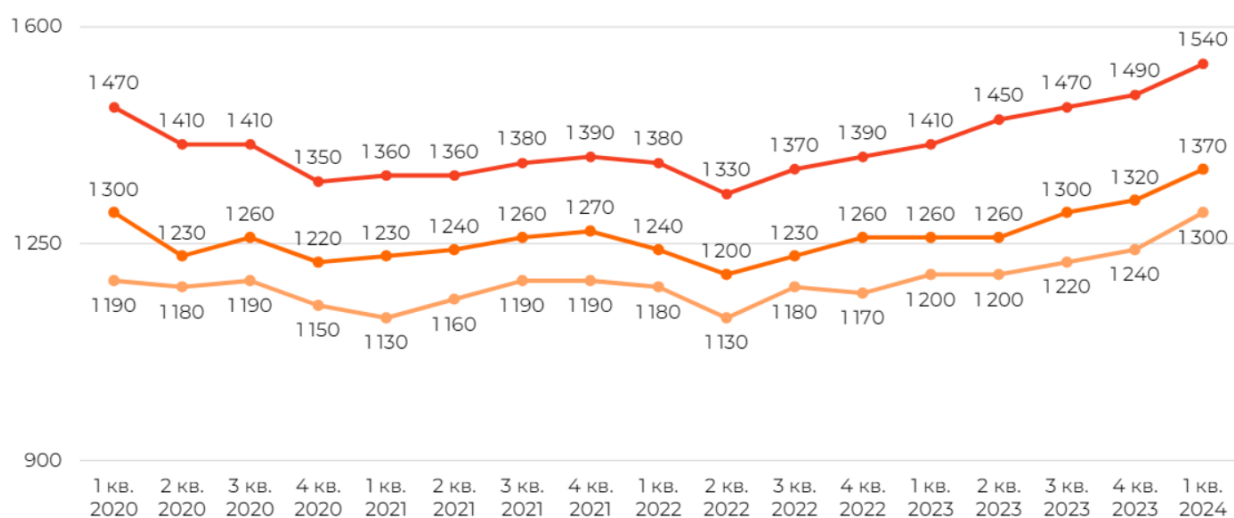


Рисунок 1.2 – Средняя цена за квадратный метр в долларах

В новостройках же в первом квартале средняя цена квадратного метра в однокомнатных квартирах составила 1480 долларов, за год увеличившись на 14%. «Учитывая небольшую долю первичного рынка в общей структуре

сделок в 2023 году, средняя цена в каждом периоде может быть подвержена влиянию продаж квартир в конкретных домах или жилых комплексах, что приводит к искусственным «скачкам» цен, но не говорит о тенденциях на первичном рынке в целом», – поясняют в Кадастровом агентстве.

Самая дорогая квартира по общей цене в 1 квартале была продана за 383 тыс. долларов. Это 4-комнатная квартира площадью 194 кв.м по улице Друйской. Самый дорогой квадратный метр оказался в 4-комнатной квартире площадью 98 кв.м по улице Ратомской. Здесь «квадрат» покупателю обошелся в 3,4 тыс. долларов [7].

Первичный рынок занял 11% продаж в столице. Из всех 455 сделок 321 сделка была совершена в микрорайоне «Минск Мир».

В новостройках абсолютное большинство проданных квартир – однокомнатные (87%). Их доля увеличилась на 12 процентных пунктов. Аналитики подчеркивают, что треть из них имеют площадь более 44 квадратных метров, то есть потенциально могут использоваться как двухкомнатные квартиры. На вторичном рынке доля однокомнатных квартир составила 41%.

В целом квартиры до 35 кв. м. занимают 32% в продажах, и их доля выросла по сравнению с предыдущим кварталом.

Большинство квартир, а именно 58% покупались в новых домах, построенных после 2010 года. Популярны также дома 1960-70 годов – на них пришлось 19% продаж.

Несмотря на изменения в экономической ситуации и социальных условиях, спрос на жилье в Республике Беларусь остается стабильным. Это связано с повышением уровня жизни, улучшением доступности ипотечного кредитования и демографическими факторами.

Постоянное развитие инфраструктуры, включая транспортную, коммунальную и социальную, оказывает существенное влияние на рынок недвижимости. Развитие новых районов и инфраструктурных объектов способствует повышению привлекательности определенных районов для жильцов.

В последние годы отмечается устойчивый рост цен на жилье в различных сегментах рынка, как в новостройках, так и на вторичном рынке. Этот рост цен обусловлен как увеличением спроса, так и увеличением стоимости строительных материалов и услуг.

Параллельно с ростом спроса на жилье наблюдается и увеличение предложения, особенно в сегменте новостроек. За счет строительства новых

объектов и реконструкции старых зданий предложение на рынке недвижимости постоянно расширяется.

В последние годы растет интерес к жилью в региональных городах Республики Беларусь. Это связано с развитием инфраструктуры и улучшением условий жизни в таких городах, что делает их привлекательными для покупателей.

Государственная поддержка жилищного строительства и ипотечного кредитования играет значительную роль в стимулировании развития рынка недвижимости в Республике Беларусь. Различные программы государственной поддержки способствуют увеличению доступности жилья для населения.

В целом, рынок недвижимости в Республике Беларусь демонстрирует стабильные тенденции роста, поддерживаемые различными факторами, и предоставляет широкие возможности как для покупателей, так и для инвесторов.

## **2 АНАЛИЗ БИЗНЕС-ПРОЦЕССОВ, СВЯЗАННЫХ С ПОКУПКОЙ КВАРТИРЫ**

### **2.1 Описание предприятия ОАО «Лидагропроммаш»**

ОАО "Лидагропроммаш" – это крупное белорусское предприятие, специализирующееся на производстве сельскохозяйственной техники и оборудования. Основанное в городе Лида, предприятие имеет долгую историю и значительный опыт в области машиностроения, что позволяет ему успешно конкурировать как на внутреннем, так и на внешнем рынках.

Его основными направлениями являются:

- производство сельскохозяйственной техники;
- производство оборудования для животноводства;
- разработка и производство запчастей и комплектующих.

Сельскохозяйственная техника включает в себя почвообрабатывающие машины, которые состоят из плуга, культиватора, и др., посевные машины, уборочную технику (комбайны, пресс-подборщики).

Оборудование для животноводства состоит из машин для транспортировки и раздачи кормов, а также различного оборудования для молочного животноводства.

Разработка и производство запчастей и комплектующих происходит с целью обслуживания и модернизации существующей сельскохозяйственной техники.

Данное предприятие имеет ряд патентов на уникальные разработки и инновационные решения в области сельскохозяйственного машиностроения. Также продукция предприятия сертифицирована по международным стандартам качества, таким как ISO 9001, что подтверждает её надежность и высокое качество.

Если говорить о международном рынке, то предприятие активно сотрудничает с партнерами из стран СНГ, Европы, Азии и других регионов, экспортируя свою продукцию в десятки стран мира. А также участвует в международных выставках и форумах, где демонстрирует свои достижения и новинки в области сельскохозяйственной техники.

Как было указано ранее, предприятие не стоит на месте ОАО "Лидагропроммаш" постоянно инвестирует в научные исследования и разработки, сотрудничая с ведущими научно-исследовательскими институтами и университетами, и регулярно внедряет передовые технологии



и автоматизированные системы управления производственными процессами для повышения эффективности и качества выпускаемой продукции.

Предприятие активно участвует в социальных проектах, поддерживает местное сообщество и спортивные мероприятия, внедряет экологически чистые технологии и решения, направленные на снижение негативного воздействия на окружающую среду.

ОАО "Лидагропроммаш" – это динамично развивающееся предприятие, сочетающее традиции и инновации для создания высококачественной и надежной сельскохозяйственной техники, способствующей развитию аграрного сектора.

## **2.2 Анализ состояния изучаемого процесса с целью выявления недостатков и обоснования целесообразности автоматизации процесса**

Проблема, стоящая перед потенциальными покупателями жилья в современном быстро развивающемся мире, заключается в сложности и затруднительности процесса выбора и принятия решения о покупке квартиры. Эта проблема обусловлена несколькими ключевыми факторами:

1 Большие объемы данных. Потенциальные покупатели сталкиваются с огромным количеством информации о рынке недвижимости, включая данные о ценах, характеристиках квартир, состоянии инфраструктуры и благоустройства районов, транспортной доступности и других важных аспектах. Обработка и анализ этих данных вручную требует значительных временных и умственных ресурсов.

2 Сложность сравнения и оценки вариантов. Каждый потенциальный покупатель имеет уникальные критерии и предпочтения при выборе квартиры. Сравнение и оценка различных вариантов по множеству критериев является сложной задачей.

3 Риск ошибок и субъективность. Принятие решений на основе интуиции или ограниченного анализа данных может привести к ошибкам и неудовлетворительным результатам.

Как было указано выше одной из основных проблем при выборе квартиры является необходимость в поиске подходящих вариантов среди большого количества объявлений. Ведь как известно, при большом выборе теряется фокус и невозможно определить лучший [8]. Это влияет на настроение и качество принимаемых решений.

После нахождения интересных вариантов квартир необходимо провести анализ и сравнение их параметров таких как площадь, расположение, состояние и т.д. Это является довольно трудоёмким и мыслезатратным процессом, прибавляем ещё то, что нужно анализировать не одну квартиру а сравнивать её с несколькими и держать в голове информации о каждой из них, и проверять лучше ли текущая квартира предыдущей и проблемой является, что каждый покупатель имеет свои индивидуальные предпочтения и потребности, которые не всегда учитываются при поиске жилья. Даже при наличии некоторых фильтров на сайте учесть все предпочтений покупателя невозможно.

Одним из решений данных проблем могут стать экспертные оценки. Эксперты в области недвижимости обладают глубокими знаниями о рынке, ценообразовании, тенденциях и других важных аспектах. Эти знания и опыт интегрируются в систему, чтобы предоставлять более точные и обоснованные рекомендации. Некоторые аспекты недвижимости, трудно количественно оценить. Эксперты могут предоставить качественные оценки, основанные на их профессиональных знаниях и интуиции. Эксперты могут выявлять потенциальные риски, связанные с покупкой определенного объекта недвижимости.

Хотя использование экспертных имеет много преимуществ, существует ряд недостатков и ограничений, которые необходимо учитывать:

1 Личная предвзятость экспертов. Эксперты могут быть подвержены личным предвзятостям или предпочтениям, которые могут повлиять на их оценки и рекомендации.

2 Вариативность оценок. Разные эксперты могут давать разные оценки одному и тому же объекту недвижимости, что может привести к противоречивым рекомендациям.

3 Узкая специализация. Эксперты могут обладать глубокими знаниями в определенных областях, но их компетенции могут быть ограничены в других аспектах, что может привести к неполным или односторонним оценкам.

4 Необновленные знания. Экспертные знания могут устаревать со временем, особенно в быстро меняющихся рынках недвижимости, что требует регулярного обновления информации и моделей.

Из вышеуказанных проблем можно сделать вывод, что главным недостатком существующих систем являются высокие временные и ресурсные затраты. Таким образом, проблема заключается в необходимости создания эффективного инструмента, который сможет облегчить процесс выбора и

принятия решения о покупке квартиры, снизить временные и умственные затраты на анализ данных.

Решением этих проблем может стать автоматизированная система поддержки принятия решений о приобретении квартиры, основанная на экспертных оценках и современных методах анализа данных. Система будет учитывать предпочтения пользователя и на основе этого, предлагать персонализированные рекомендации по подходящим вариантам жилья.

### **2.3 Описание процесса покупки квартиры as-is**

Для визуального представления нашего процесса провизуализируем его при помощи методологии IDEF0. Методология IDEF0 предписывает построение иерархической системы диаграмм – единичных описаний фрагментов системы. Сначала проводится описание системы в целом и ее взаимодействия с окружающим миром (контекстная диаграмма), после чего проводится функциональная декомпозиция – система разбивается на подсистемы, и каждая подсистема описывается отдельно (диаграммы декомпозиции). Затем каждая подсистема разбивается на более мелкие и так далее до достижения нужной степени подробности. Каждая IDEF0 диаграмма содержит блоки и дуги. Блоки изображают функции моделируемой системы. Дуги связывают блоки вместе и отображают взаимодействия и взаимосвязи между ними. Функциональные блоки (работы) на диаграммах изображаются прямоугольниками, означающие поименованные процессы, функции или задачи, которые происходят в течение определенного времени и имеют распознаваемые результаты. Имя работы должно быть выражено отглагольным существительным, обозначающим действие. Каждая сторона блока имеет особое, вполне определенное назначение. Левая сторона блока предназначена для входов, верхняя – для управления, правая – для выходов, нижняя – для механизмов (рисунок 2.1).

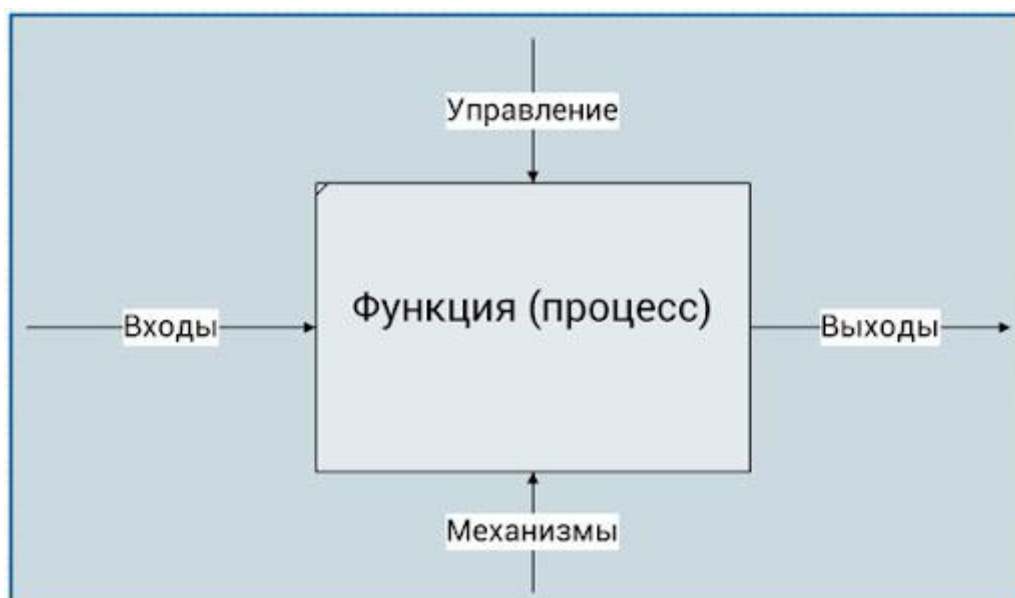


Рисунок 2.1 – Обязательные элементы функционального блока IDEF0.

Для наглядного представления процесса покупки квартиры была разработана графическая схема AS-IS на основе стандарта IDEF0. На этой схеме изображён процесс покупки квартиры до его автоматизации.

Начнём наш процесс с контекстной диаграммы (рисунок 2.2). На ней изображён процесс «Купить квартиру». Входным параметром является запрос покупателя, а на выходе купленная квартира. Механизмами в данном процессе являются: продавец квартиры, покупатель, банк, нотариус, агент по недвижимости, сайты с квартирами. Ну и за управление отвечают: нормативные документы, правовые нормы, закон о защите прав потребителей.

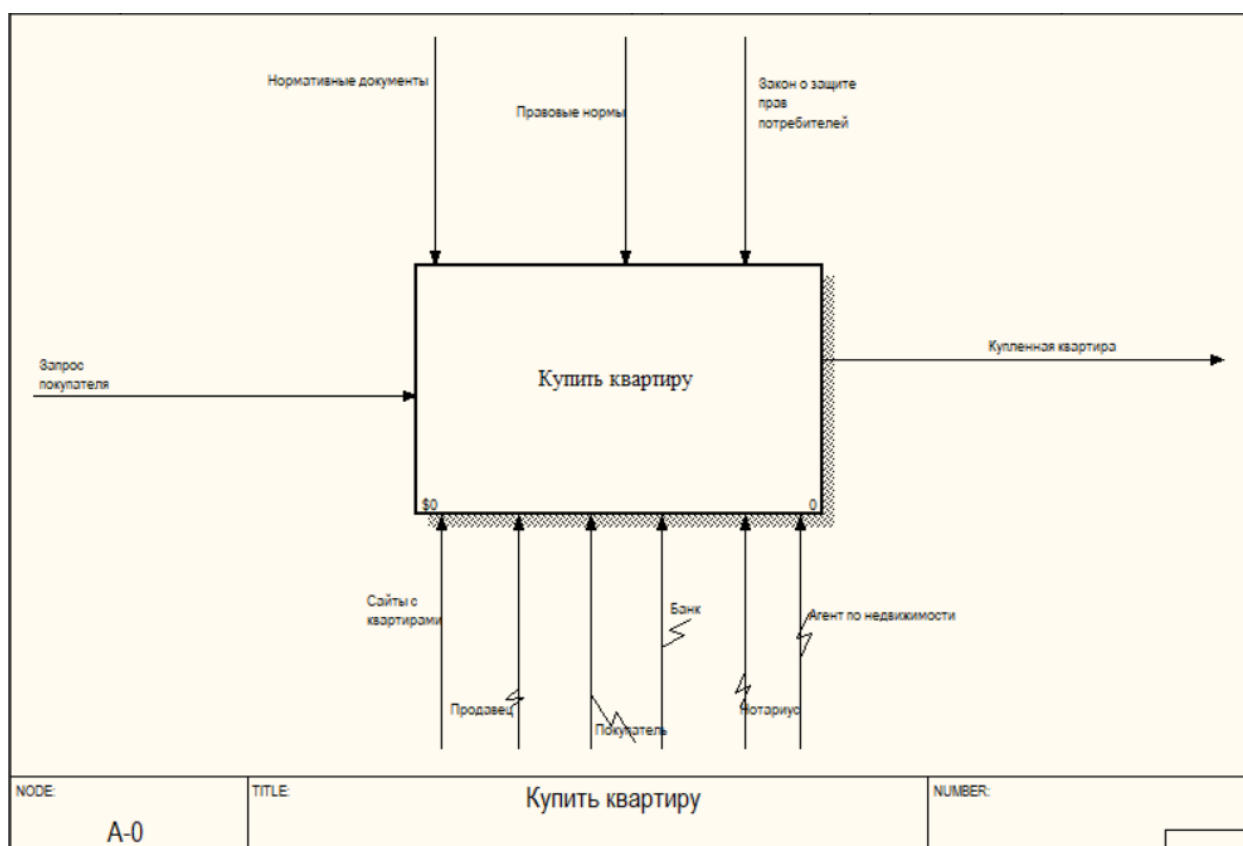


Рисунок 2.2 – Контекстная диаграмма

В процесс покупки квартиры входит несколько частей (рисунок 2.3).

Первая часть – это определение критериев, параметров, по которым квартиры будут избираться. На нём потенциальный покупатель определяет, что для него важно, а что нет.

Вторая часть – осмотр вариантов. На этом этапе потенциальный покупатель, из большого множества квартир выбирает некий вариант основываясь на своих предпочтениях и важных ему критериях, выбранных на предыдущем шаге.

Третья часть – сбор документов. На этом этапе не обойтись без соответствующих людей как нотариус, который поможет собрать нужные документы и заверит сделку в её юридической чистоте.

Четвёртая часть – совершение сделки. На данном этапе уже подписываются все ранее собранные документы и договора, перечисляются деньги со счёта покупателя на счёт продавца и происходит передача собственности.

Пятая часть – регистрация договора в национальном кадастровом агентстве.

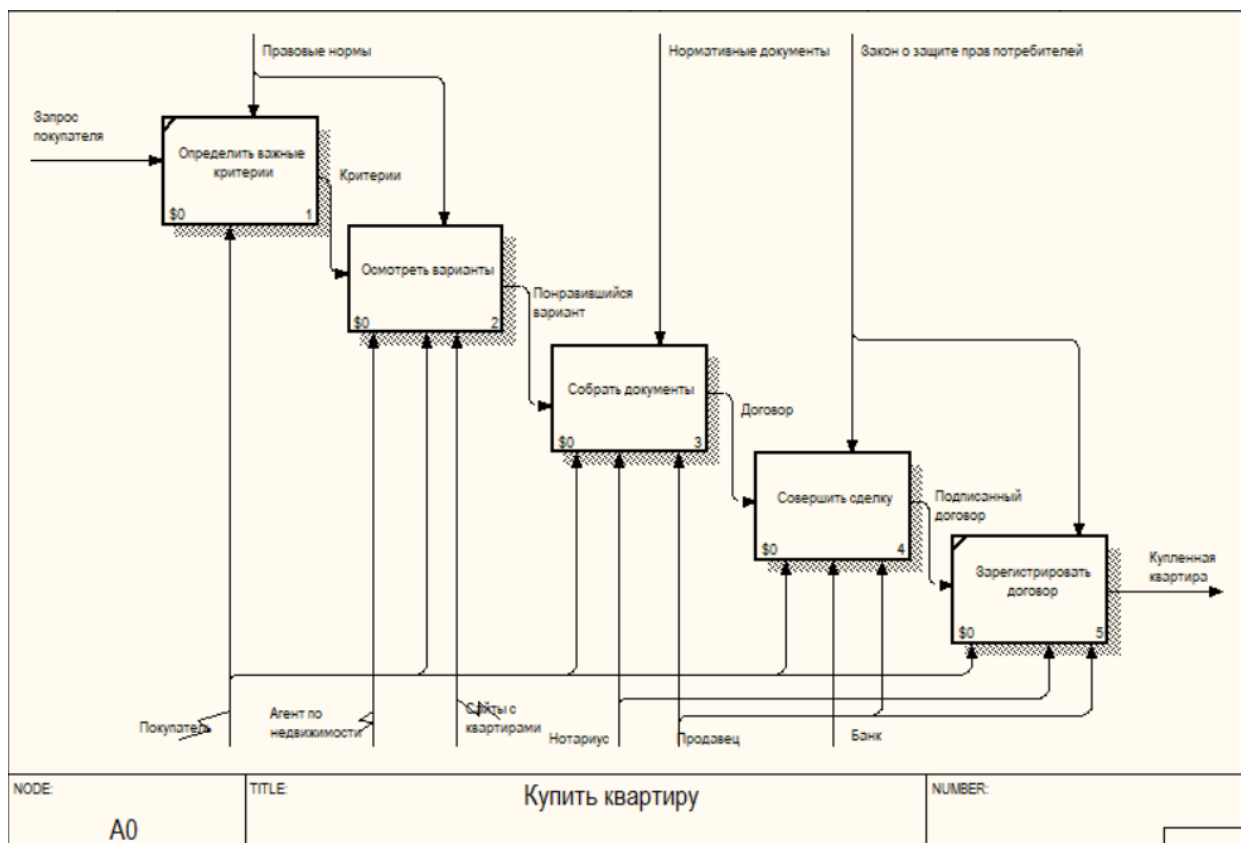


Рисунок 2.3 – Декомпозиция контекстной диаграммы

При чуть более детальном изучении процесса осмотра различных вариантов квартир можем заметить, что он состоит ещё из нескольких процессов. На рисунке 2.4 изображена декомпозиция процесса «Осмотреть квартиры». При детальном анализе данного процесса можно сказать, что он не так и сложен и состоит всего из трёх частей: осмотр вариантов, предложенных агентом, осмотр квартир на сайтах, определение понравившегося варианта. Но не стоит недооценивать количество шагов с их времязатратами, так как на осмотр, подборку и выборку уходит значительное количество времени. И бывают случаи, что из-за длительного процесса поиска подходящего варианта потенциальный покупатель вообще отказывается от идеи покупки квартиры. Поэтому здесь важно предоставить именно персонализированные варианты для простоты подбора и уменьшению временных затрат.

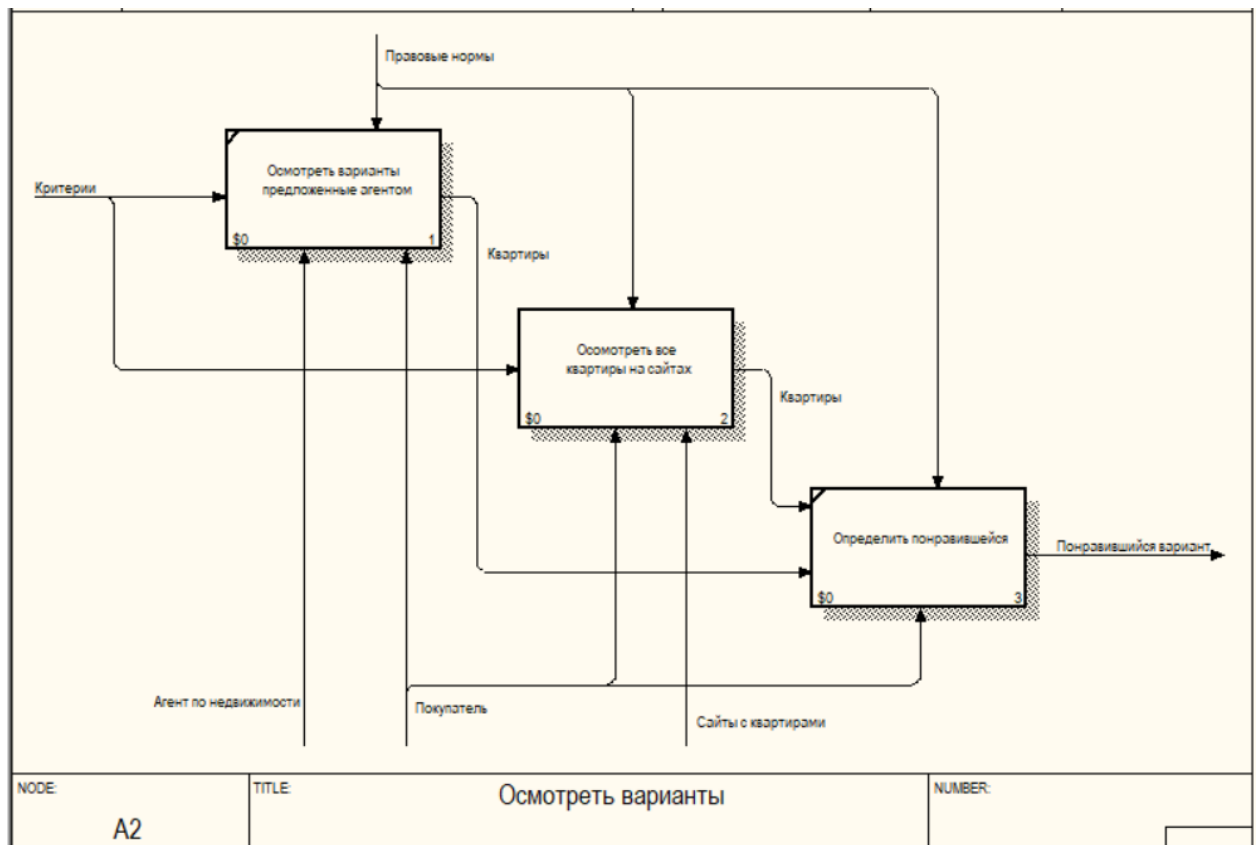


Рисунок 2.4 – Декомпозиция блока «Осмотреть варианты»

Также, если капнуть немного глубже в процесс осмотра квартир на сайтах, то можно увидеть, что он довольно тривиален (рисунок 2.5). Но также, не стоит забывать об упомянутых ранее факторах времени, ведь процесс поиска подходящих квартир, даже с учётом применения некоторых фильтров, может занять длительный промежуток времени. В связи с этим стоит задуматься об автоматизации данного процесса, что будет представлено позже в описании процесса as to be.

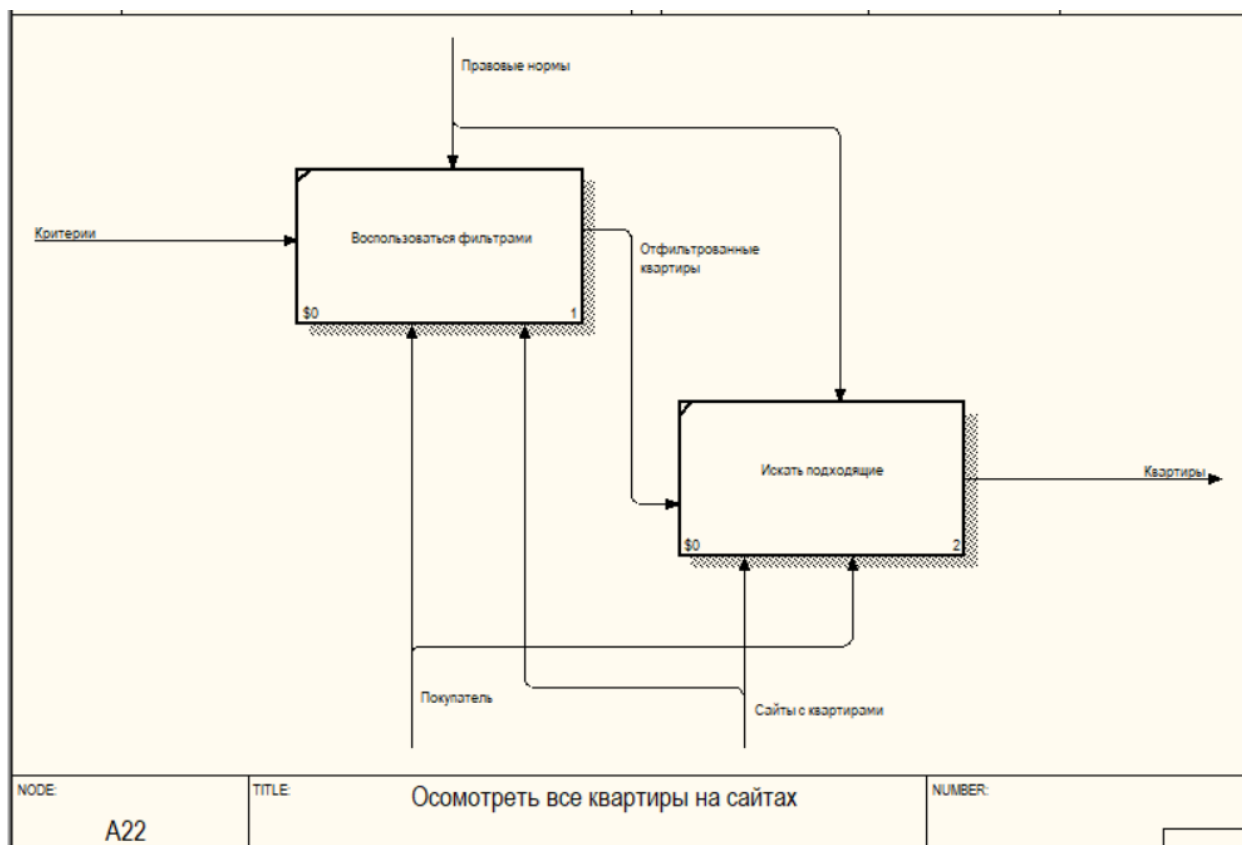


Рисунок 2.5 – Декомпозиция блока «Осмотреть все квартиры на сайтах»

Продолжим исследование процесса покупки квартиры с более детального рассмотрения блока «Собрать документы» (рисунок 2.6). Как и было упомянуто ранее, здесь не обойтись без людей, которые разбираются в юриспруденции. Чтобы сделка прошла без проблем, а именно лишних трат времени и возможных проблем с законом, из-за неправильно составленных документов, нужно обратиться за помощью к нотариусу, который разбирается в этом вопросе и поможет с ним. Сделка будет юридически чиста и покупатель, также как и продавец, выполнят задуманное.



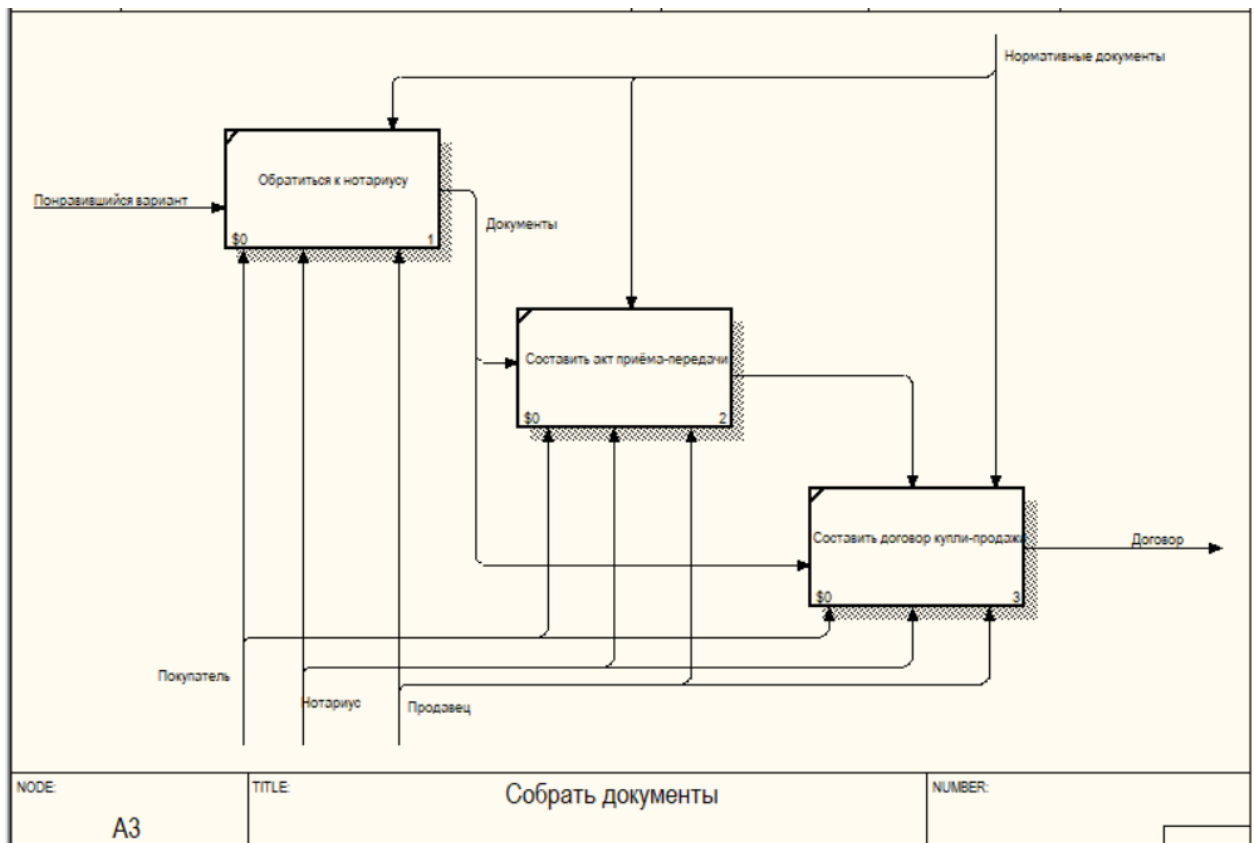


Рисунок 2.6 – Декомпозиция блока «Собрать документы»

После сбора документов идёт уже процесс совершения сделки (рисунок 2.7). Здесь происходит передача средств от покупателя к продавцу и подписываются ранее сделанные документы о передачи собственности. При оплате также участвует банк, чтобы деньги, переданные от покупателя к продавцу были законны (рисунок 2.8).

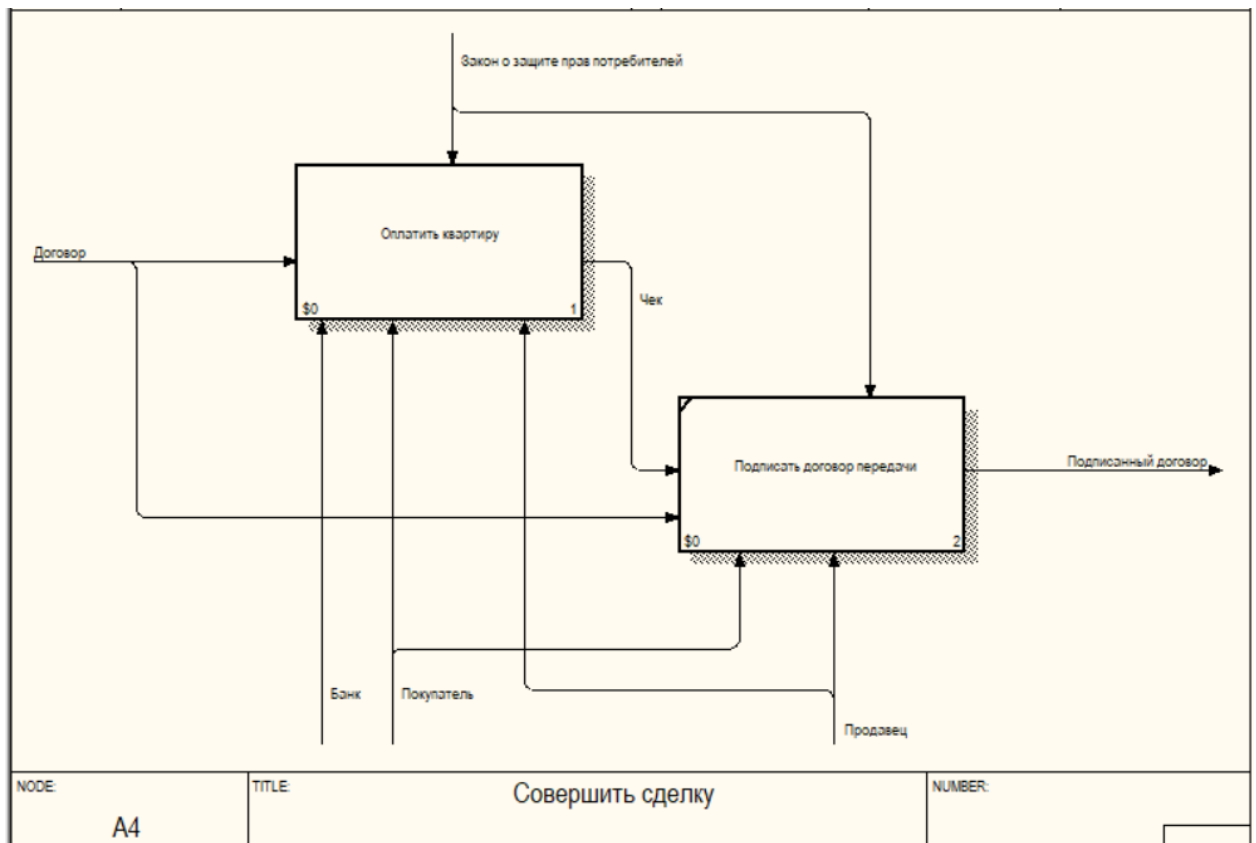


Рисунок 2.7 – Декомпозиция блока «Совершить сделку»

Процесс, представленный выше довольно прост и в то же время сложен, так как процесс перевода со счёта на счёт, тем более довольно больших сумм весьма сложен, но этот процесс выходит за рамки обсуждаемой темы и нам, как пользователям банковских систем стоит довериться абстракциям и надёжности банковских систем. Чуть более детально данный процесс описан на рисунке 2.8 но это также, не вдаваясь в подробности работы банковских систем. Важно чтобы деньги убыли из одного места, от покупателя, и прибыли в другое место, к продавцу, в равных количествах.

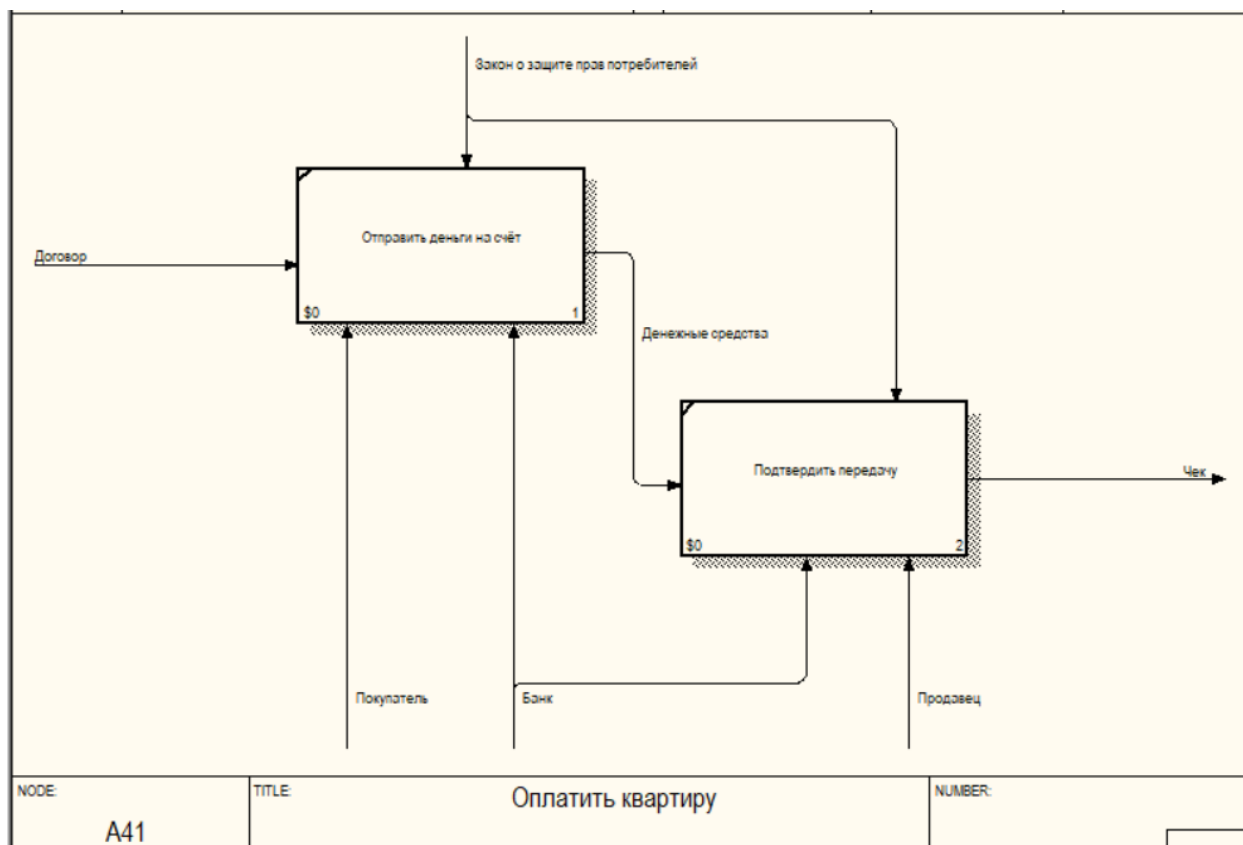


Рисунок 2.8– Декомпозиция блока «Оплатить квартиру»

Таким образом был представлен процесс покупки квартиры как он существует сейчас. Далее будет представлено, как будет работать данный процесс с учётом его автоматизации.

## 2.4 Описание процесса покупки квартиры as to be

Как было рассмотрено ранее, процесс осмотра занимает длительное время. В связи с этим идея ускорения данного процесса выглядит весьма разумной. Поэтому была предложена система «умного поиска» (рисунок 2.9). Это система замещает обычные фильтры более продвинутыми, которые будут основываться не только на характеристиках квартиры, а и на оценках характеристик данной квартиры другими пользователями.

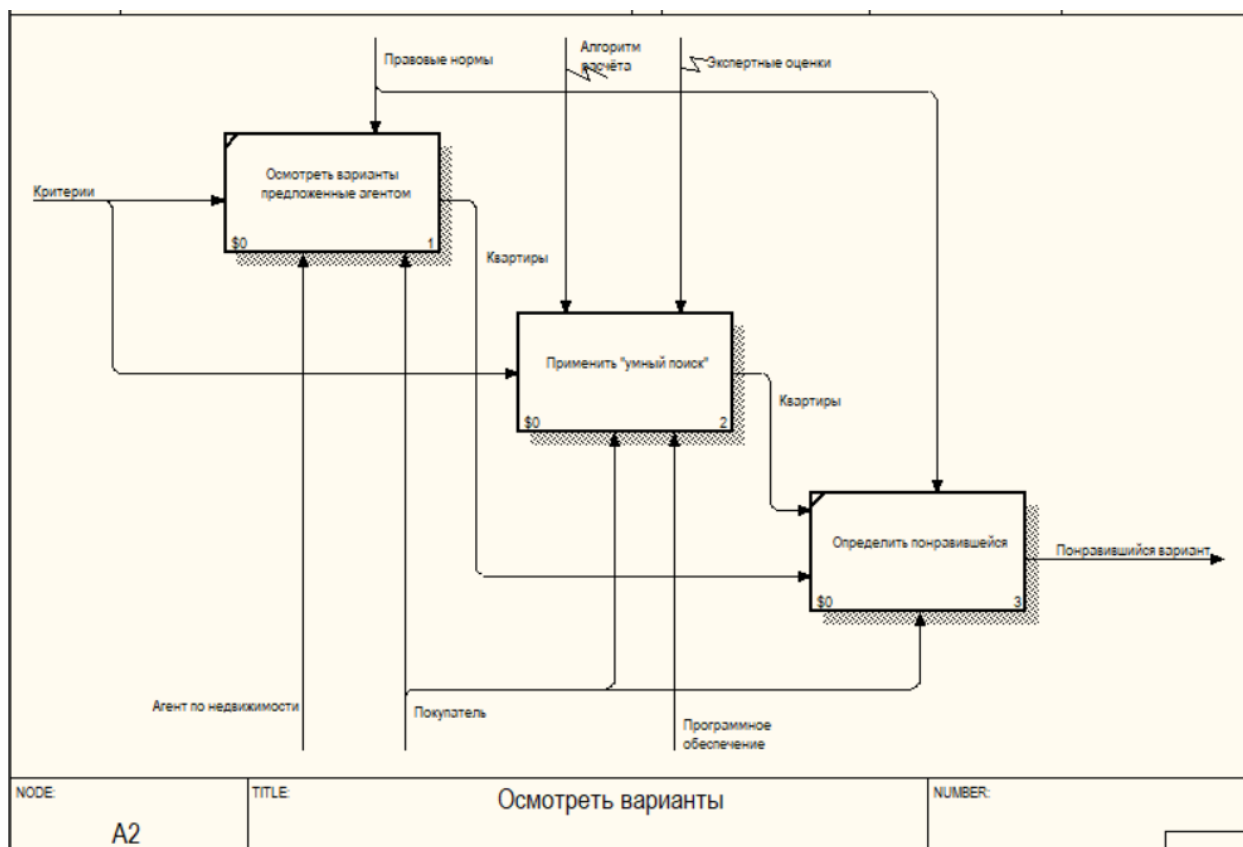


Рисунок 2.9 – Автоматизация процесса «Осмотреть варианты»

На рисунке 2.10 показано как будет выглядеть процесс поиска подходящего жилья при автоматизации поиска. Пользователю достаточно просто ввести важность параметров, а всё остальное сделает система и выдаст наиболее подходящие квартиры, основываясь на важности параметров, предоставленных пользователем. Абсолютно всё учесть невозможно, но для более точного определения нужд человека система будет учитывать как можно больше параметров.

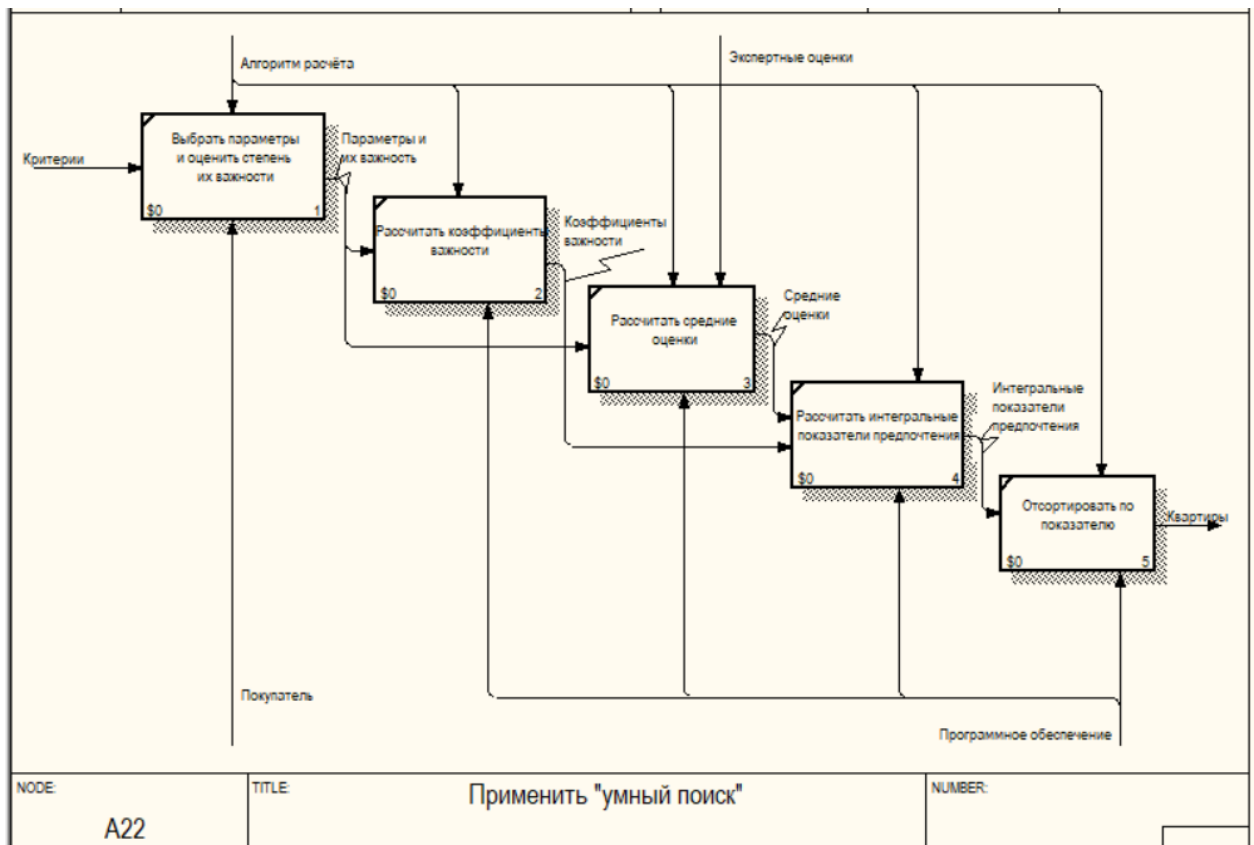


Рисунок 2.10 – Декомпозиция блока «Применить умный поиск»

Таким образом и будет устроена автоматизации процесса покупки квартиры. С применением новейших технологий и методов анализа.

## 2.5 Обзор существующих систем

Сравнительный анализ аналогов, позволяет выявить особенности местных платформ и определить, какие инновации, направления и усовершенствования может предложить проектируемая система для максимального удовлетворения потребностей пользователей. Данный анализ выделяет ключевые характеристики, функциональные особенности и преимущества существующих решений, а также выявляет области, где разрабатываемая система может предложить улучшения.

На белорусском рынке существует несколько аналогов, ориентированных на предоставление информации о недвижимости. Важно учитывать особенности этого рынка, местные стандарты и потребности пользователей. Рассмотрим несколько платформ.

«Realt.by», в нём выделяются следующие преимущества:

- крупнейший в Беларуси портал по недвижимости;

- обширная база данных с предложениями по продаже и аренде;
- разнообразные фильтры для удобного поиска.

Также, как и у всех, у неё есть и недостатки: некоторые пользователи отмечают несовершенства в работе фильтров и отсутствие некоторых продвинутых функций. На рисунке 2.11 изображена страница с недвижимостью Realt.

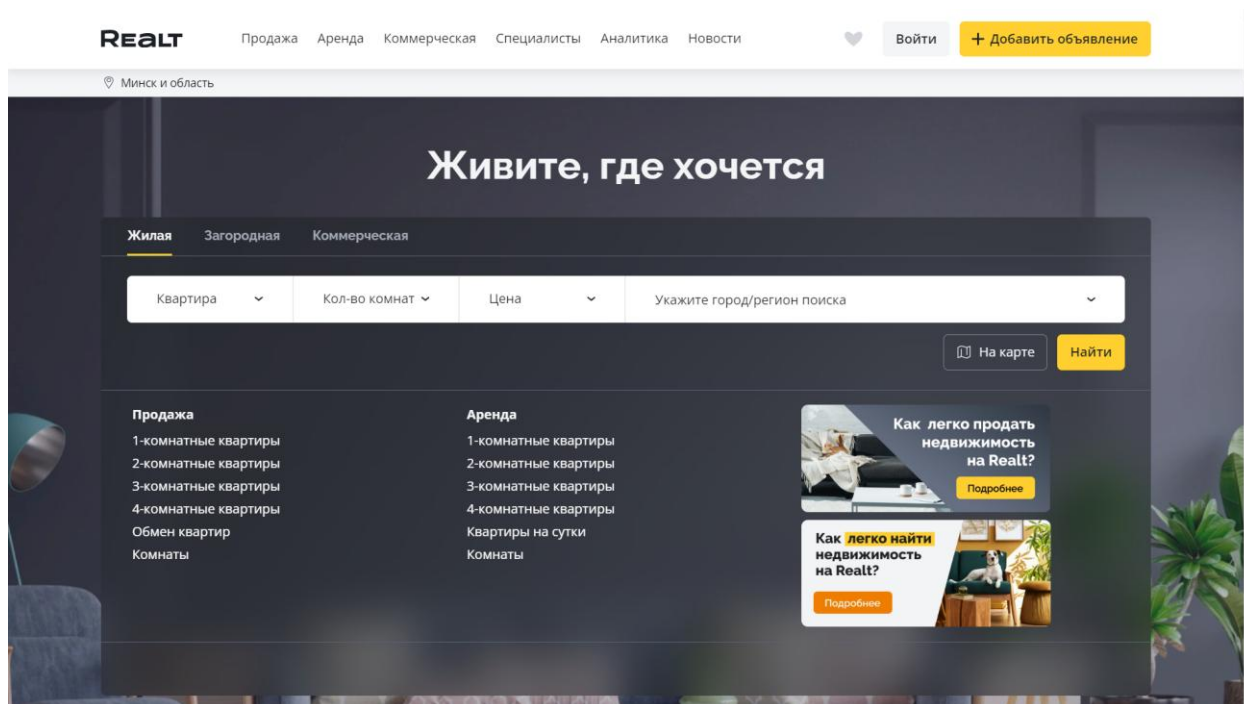


Рисунок 2.11 – Страница с недвижимостью Realt

«Onliner Недвижимость», в нём выделяют следующие преимущества:

- интеграция с крупнейшим технологическим порталом Onliner;
- подробная информация о жилье и возможность связаться с продавцом.

Из недостатков это ограниченный функционал в сравнении с некоторыми мировыми аналогами и не всегда актуальная информация о доступных объектах. На рисунке 2.12 изображена страница с недвижимостью Onliner.

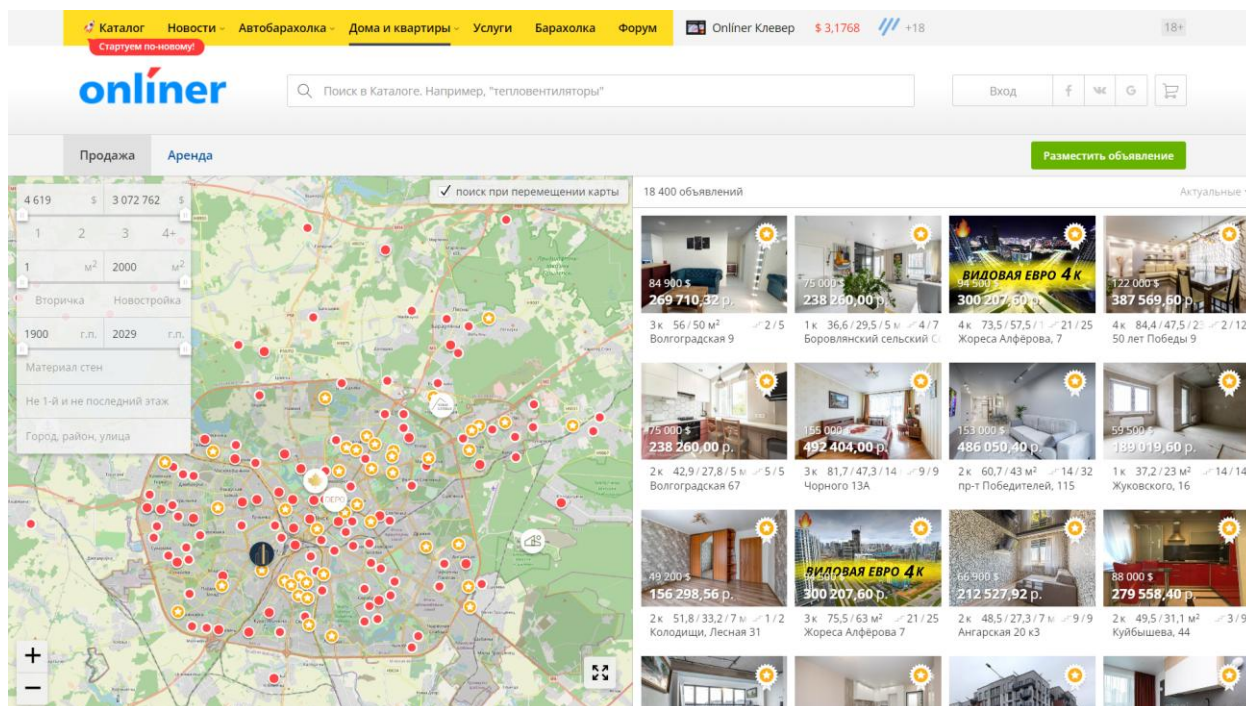


Рисунок 2.12 – Страница с недвижимостью Onliner

«Kufar Недвижимость», основные преимущества в данном сервисе это:

- большое количество предложений от частных лиц;
- удобная система фильтрации и сортировки.

Из недостатков можно выделить меньшую детализацию и фотографии объектов по сравнению с профессиональными агентствами и ограниченный поиск в нескольких крупных городах. На рисунке 2.13 изображена страница с недвижимостью Kufar.



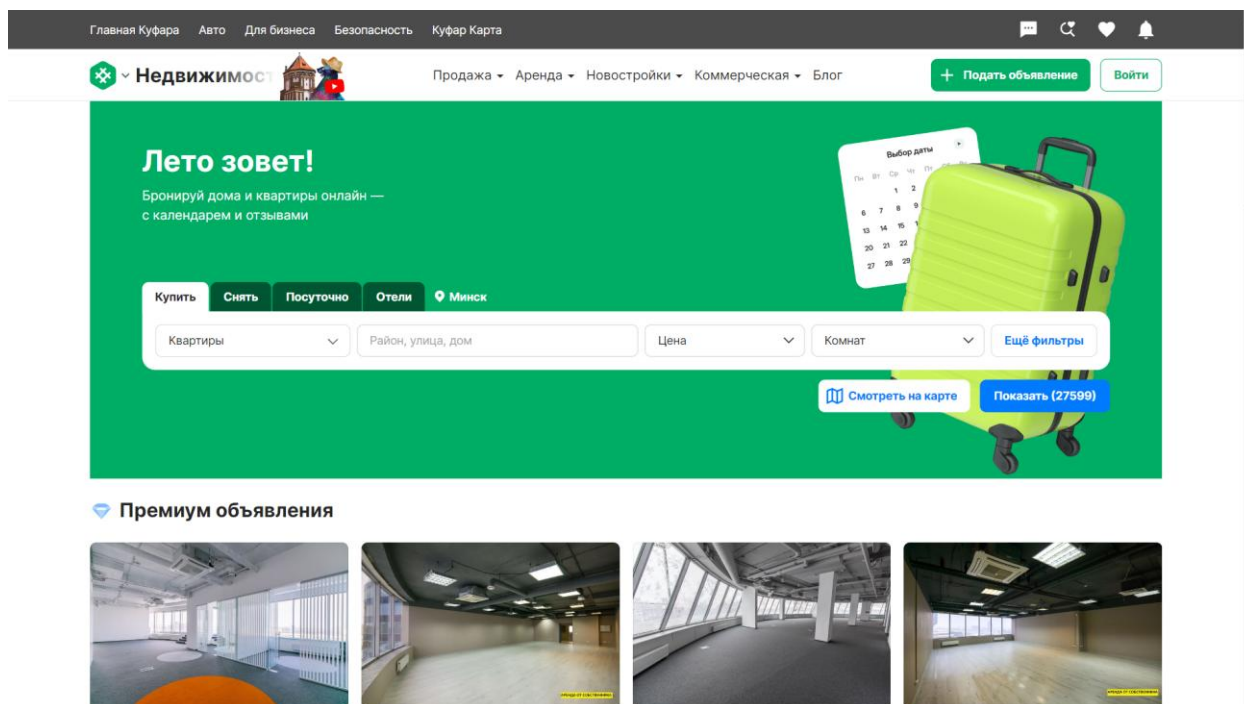


Рисунок 2.13 – Страница с недвижимостью Kufar

В результате проведенного анализа литературных исследований и программных решений, предоставленных в сфере автоматизированных систем поддержки принятия решений о приобретении жилья, было выявлено ряд важных наблюдений и трендов.

Во-первых, обзор данной области дал представление о современных тенденциях и проблемах, с которыми сталкиваются пользователи. Это позволило сформировать фундаментальное понимание основных аспектов принятия решений в контексте недвижимости.

Во-вторых, рассмотрение существующих программных решений выявило преимущества и ограничения различных платформ.

Обобщая полученные данные, можно заключить, что разработка автоматизированной системы поддержки принятия решений о приобретении квартиры должна учитывать требования к точности, персонализации и интеграции современных технологий.

На основе проведенного анализа будет разработана концепция автоматизированной системы, учитывающая опыт и лучшие практики, выделенные в ходе изучения литературы и анализа программных решений.



## 2.6 Основные подходы к поддержке принятия решений о приобретении квартиры

Системы поддержки принятия решений используются для обеспечения задач различных уровней управления и помогают людям принимать решения в сложных ситуациях. Совершенствование инструментов анализа ситуаций и принятия решений приводят к появлению новых подходов в организации деятельности, которые должны соответствовать новым условиям и характеризуются постоянным повышением динамичности процессов, сложностью вариантов принимаемых решений, высоким уровнем неопределенности исследуемых явлений. Существенно увеличились возможные риски, масштабы и размеры потерь, которые могут возникнуть при принятии недостаточно обоснованных решений. Как следствие, увеличилась ответственность человека за принятие «неправильного» решения, возросли трудности, связанные с его нахождением, преодолеть которые невозможно без использования новых методов анализа информации и моделирования ситуаций.

Современные тенденции развития методологии поддержки принятия решений связаны с интеллектуальным анализом данных и искусственным интеллектом. С каждым годом появляется все больше сфер деятельности, в которых внедряются подходы, ориентированные на работу с большими массивами разнородных данных и извлечением знаний, созданием систем управления на основе данных.

Умение работать с разными математическими методами и моделями поддержки принятия решений является одной из наиболее перспективных и востребованных компетенций ближайшего времени [9].

Задача принятия решения состоит в формировании множества возможных вариантов, обеспечивающих разрешение проблемной ситуации при существующих ограничениях, и выделении среди этих вариантов одного лучшего или нескольких предпочтительных вариантов, удовлетворяющих предъявляемым к ним требованиям. Формально задачу принятия решения  $D$  можно записать в следующем обобщенном виде [10]:

$$D = \{F, A, X, G, P\}, \quad (2.1)$$

где  $F$  – формулировка задачи принятия решения, которая включает в себя содержательное описание стоящей проблемы и при необходимости ее модельное представление, определение цели

или целей, которые должны быть достигнуты, а также требования к виду окончательного результата;

- А – совокупность возможных вариантов (альтернатив), из которых производится выбор. Это могут быть реально существующие варианты, в качестве которых в зависимости от контекста задачи выступают объекты, кандидаты, способы достижения цели, действия, решения и т.п., либо гипотетическое множество всех теоретически возможных вариантов, которое может быть даже бесконечным. Выбор возникает только тогда, когда имеется не менее двух возможных вариантов решения проблемы;
- Х – совокупность признаков (атрибутов, параметров), описывающих варианты и их отличительные особенности. В качестве признаков выступают, во-первых, объективные показатели, которые характеризуют те или иные свойства, присущие вариантам, и которые, как правило, можно измерить; во-вторых, субъективные оценки, которые обычно даются по специально отобраным или сконструированным критериям, отражающим важные для участников выбора черты вариантов;
- Г – совокупность условий, ограничивающих область допустимых вариантов решения задачи. Ограничения могут быть описаны как содержательным образом, так и заданы в виде некоторых формальных требований к вариантам и/или их признакам. Например, это могут быть ограничения на значения какого-либо признака или различная степень характерности (выраженности) признака для тех или иных вариантов, или невозможность одновременного сочетания определенных значений признаков для реально существующих вариантов;
- Р – предпочтения одного или нескольких ЛПР, которые служат основой для оценки и сравнения возможных вариантов решения проблемы, отбора допустимых вариантов и поиска наилучшего или приемлемого варианта.

До последнего времени самым используемым способом снятия неопределенности было привлечение экспертов или руководителей, субъективное мнение которых становилось основой для определения критериев и выбора альтернатив.

В настоящее время становятся всё более популярными алгоритмы искусственного интеллекта.

Остановимся на том как будет работать система с использованием экспертных оценок.

Интегральный показатель предпочтения рассчитывается по формуле:

$$S = \sum_{i=1}^n K_i \cdot X_i, \quad (2.2)$$

где  $K_i$  – коэффициент важности  $i$ -го параметра;

$X_i$  – средняя оценка  $i$ -го параметра.

Коэффициент важности рассчитывается по формуле:

$$K = \frac{k_i}{\sum_{i=1}^n k_i}, \quad (2.3)$$

где  $k_i$  – важность  $i$ -го параметра.

Теперь перейдём к примеру на числах. Допустим у нас есть некоторая квартира с параметрами и их значениями, а также несколько пользователей, которые могут ставить оценки (таблица 2.1).

Таблица 2.1 – Оценки одной квартиры

Параметры	Значения	Пользователь №1	Пользователь №2	Среднее
Близость к метро	10 мин	2	3	2,5
Близость к ТЦ	5 мин	4	5	4,5
Общая площадь	60 м <sup>2</sup>	5	4	4,5
Район	Советский	2	4	3
Цена	70000\$	5	3	4

В данном примере используется 5 параметров, оценки которых могут варьироваться от единицы до пяти, и 2 пользователя, но в реальности параметров, как и пользователей намного больше.

Далее некий иной пользователь при поиске соответствующей квартиры вводит важность параметров, которая может варьироваться от нуля до четырёх (таблица 2.2).

Таблица 2.2 – Важность параметров

Параметр	Важность	Коэффициент важности
Близость к метро	4	0,25
Близость к ТЦ	0	0
Общая площадь	5	0,3125
Район	3	0,1875
Цена	4	0,25

Таким образом интегральный показатель предпочтения составит:

$$S = 0,25*2,5 + 0*4,5 + 0,3125*4,5 + 0,1875*3 + 0,25*4 = 3,59375.$$

И далее подобным образом рассчитывается интегральный показатель для каждой квартиры и соответственно где он больше, так квартира наиболее подходящая.

Таким образом и будет реализована система подбора квартир для пользователей.

## **3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ О ПРИОБРЕТЕНИИ КВАРТИРЫ**

### **3.1 Постановка задачи**

Основной целью работы уменьшение времязатрат на поиск квартир путём создания Web приложения.

Задача проекта состоит в том, что стоит разработать программу, которая будет отвечать всем современным тенденциям, но не будет перегружена лишней информацией.

Функции, которые могут выполнять пользователи в данном проекте:

- просмотр всех квартир;
- просмотреть отдельной квартиры;
- оценить квартиру;
- авторизоваться;
- зарегистрироваться;
- добавить квартиру в понравившиеся;
- отфильтровать квартиры;
- отсортировать квартиры;
- просмотреть последние просмотренные квартиры.

Для выполнения поставленной задачи необходимо сформировать базу данных для хранения требующейся для работы информации, а также организовать удобный для пользователя интуитивно понятный интерфейс, позволяющий работать с программой без специальных навыков.

Перечислим этапы разработки системы:

- спроектировать структуру базы данных;
- выстроить информационную модель;
- спроектировать архитектуру приложения;
- реализовать приложение на языке Java;
- протестировать и отладить работу приложения.

### **3.2 Обоснование выбора компонентов и технологий**

Для достижения поставленной задачи в Java имеется огромное количество способов и решений.

Главным преимуществом языка Java является его гибкость. Следующим достоинством является: независимость от платформы. К достоинствам также

языка относится наличие многочисленных библиотек классов и механизмов, существенно упрощающие программирование на языке Java [11].

Средой разработки данной программы является IntelliJ IDEA. IntelliJ IDEA – интегрированная среда разработки программного обеспечения для многих языков программирования, в частности Java, JavaScript, Python, разработанная компанией JetBrains. К ее достоинствам относятся эффективные инструменты минимизации ошибок (проверка и оптимизация кода, наглядное выделение и отчеты об ошибках, поиск дублирующихся фрагментов кода, идентификация зависимостей и т.д.). Среда разработки также предоставляет функцию множественного выделения, позволяющую вносить в код правки в нескольких местах одновременно. Платформа предоставляет инструменты для работы с базами данных и SQL файлами, включая удобный клиент и редактор для схемы базы данных.

Приложение будет разрабатываться на Spring Framework, а именно с использованием Spring Boot.

Для начала разберемся, что такое Spring Framework.

Spring Framework – универсальный фреймворк с открытым исходным кодом для Java-платформы.

Spring Framework может быть рассмотрен как коллекция меньших фреймворков или фреймворков во фреймворке. Большинство этих фреймворков может работать независимо друг от друга, однако они обеспечивают большую функциональность при совместном их использовании. Эти фреймворки делятся на структурные элементы типовых комплексных приложений [12].

Inversion of Control – контейнер: конфигурирование компонентов приложений и управление жизненным циклом Java-объектов.

Фреймворк аспектно-ориентированного программирования: работает с функциональностью, которая не может быть реализована возможностями объектно-ориентированного программирования на Java без потерь.

Фреймворк доступа к данным: работает с системами управления реляционными базами данных на Java-платформе, используя JDBC и ORM средства и обеспечивая решения задач, которые повторяются в большом числе Java-based environments [13].

Фреймворк управления транзакциями: координация различных API управления транзакциями и инструментарий настраиваемого управления транзакциями для объектов Java.

А теперь посмотрим, что такое Spring Boot.

Spring Boot – это полезный модуль, целью которого является упрощение создания приложений на основе Spring. Он позволяет наиболее простым способом создать web-приложение, требуя от разработчиков минимум усилий по его настройке и написанию кода.

Spring Boot обладает большим функционалом, но его наиболее значимыми особенностями являются: управление зависимостями, автоматическая конфигурация и встроенные контейнеры сервлетов .

Клиентская сторона будет реализована с использованием React.

React – это библиотека JavaScript для построения пользовательских интерфейсов [14]. Разработанная Facebook, React предоставляет декларативный и эффективный способ создания компонентов пользовательского интерфейса, которые могут быть переиспользованы и поддерживаются в виде отдельных блоков. React используется для построения динамических и отзывчивых интерфейсов во многих веб-приложениях, а также может быть интегрирован с различными библиотеками и фреймворками.

Для передачи данных между клиентом и сервером будет использоваться REST.

REST (Representational state transfer) – это стиль архитектуры программного обеспечения для распределенных систем, таких как World Wide Web, который, как правило, используется для построения веб-служб [15].

В программном средстве используется база данных PostgreSQL.

PostgreSQL — это объектно-реляционная система управления базами данных (ORDBMS), наиболее развитая из открытых СУБД в мире. Имеет открытый исходный код [16]. Среди преимуществ можно выделить: кроссплатформенность, поддержка различных типов данных, соответствие ACID, открытый исходный код.

### **3.3 Модели представления системы и их описание**

**Диаграмма вариантов использования.** Диаграмма вариантов использования (Use Case Diagram) является исходным концептуальным представлением системы в процессе ее проектирования и разработки. Данная диаграмма состоит из актёров, вариантов использования и отношений между ними. При построении диаграммы могут использоваться также общие элементы нотации: примечания и механизмы расширения.

Вариант использования, который инициализируется по запросу пользователя, представляет собой законченную последовательность действий.

Это означает, что после того, как система закончит обработку запроса актера, она должна возвратиться в состояние, в котором готова к выполнению следующих запросов.

Варианты использования могут включать в себя описание особенностей способов реализации сервиса и различных исключительных ситуаций, таких как корректная обработка ошибок системы. Связи между актерами и вариантами отображаются с использованием отношений четырех видов:

- ассоциаций;
- обобщения;
- включения (зависимость со стереотипом «include»);
- расширения (зависимость со стереотипом «extend»).

На рисунке 3.1 изображена диаграмма вариантов использования.



Рисунок 3.1 – Диаграмма вариантов использования

На ней представлено множество вариантов взаимодействия с приложением. Начнём с сортировок, пользователь может выбрать в каком



порядке будет представлена информация: дешевые квартиры, дорогие, новые или же самые популярные, данный критерий определяется по количеству просмотров. Далее идёт фильтрация данных, пользователь может комбинировать различные фильтры между собой и находить квартиры нужные ему. Также стоит учесть тот факт, что сортировка данных также может быть применена к уже отфильтрованным квартирам. Вариант использования «Умный поиск» это название того, как будет происходить поддержка принятия решения, описание работы данной функции было описано в пункте 2.5 а также на рисунке Б.1, с более подробным описанием работы в пункте 3.5. При регистрации или же авторизации пользователя ему открываются дополнительные варианты использования. При авторизации или регистрации пользователь может оценить квартиру или же изменить уже поставленные оценки. Также ему даётся доступ к функции «лайк» где он может добавить понравившуюся квартиру в отдельный список и позже просмотреть его. Этот список находится в личном кабинете пользователя, где также представлены списки оценённых квартир пользователем, а также список последних просмотренных квартир.

**Диаграмма развёртывания.** Диаграмма развёртывания представляет физическое расположение системы, показывая, на каком физическом оборудовании запускается та или иная составляющая программного обеспечения [17].

Диаграмма развёртывания представлена на рисунке 3.2

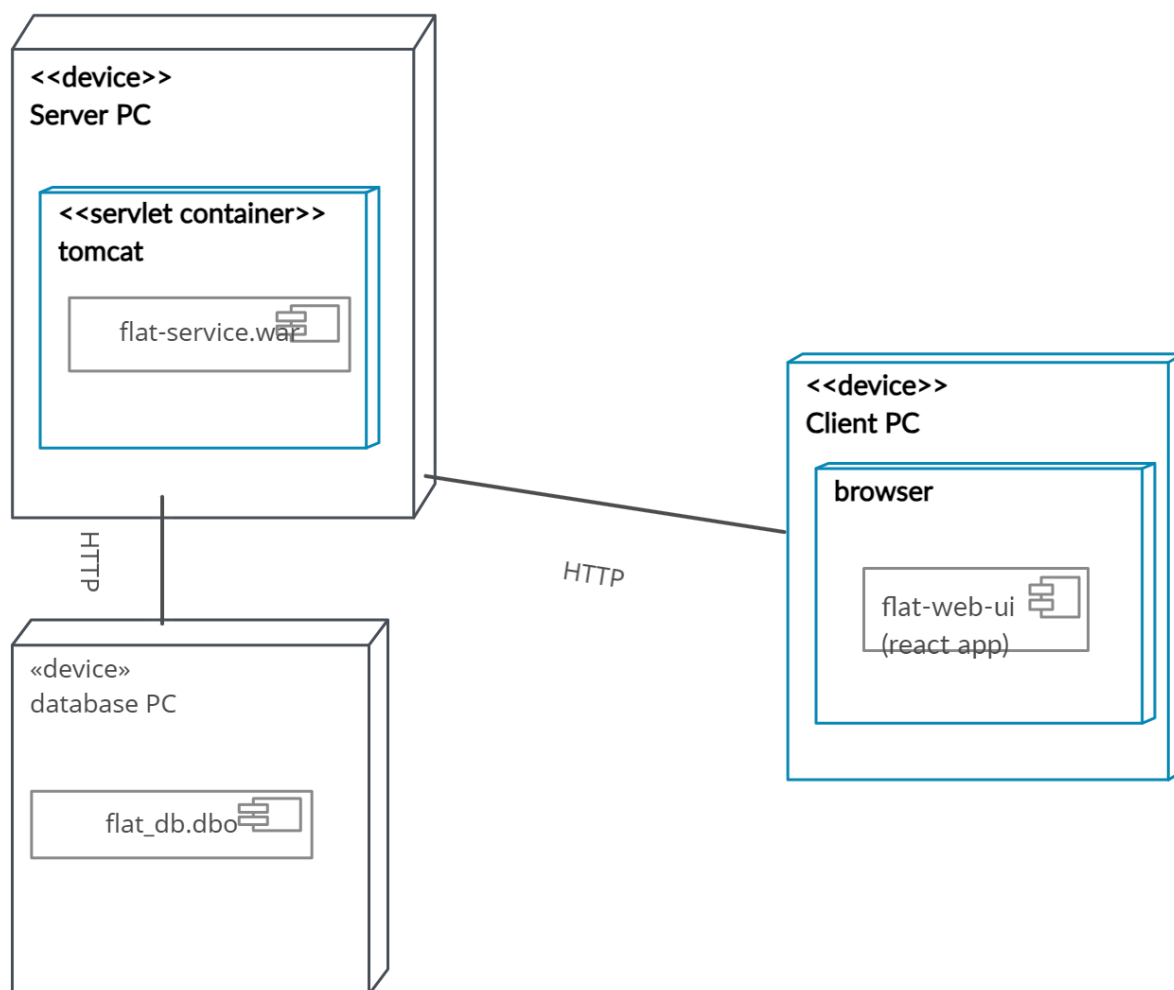


Рисунок 3.2 – Диаграмма развёртывания

Архитектура системы является клиент-серверной, то есть программный комплекс состоит из приложения клиента и приложения сервера, взаимосвязанных между собой HTTP протоколом. Одно из приложений разработаны на языке Java, поэтому для своей работы требуют установки пакета JDK, второе же написано с помощью React и некоторых фреймворков и является браузерным приложение.

Сервер связан с хранилищем данных – базой данных PostgreSQL, состоящей из 6 таблиц.

**Диаграмма классов.** Диаграмма классов – это UML-диаграмма, которая описывает систему, визуализируя различные типы объектов внутри системы и виды статических связей, которые существуют между ними. Он также иллюстрирует операции и атрибуты классов [18].

На рисунке 3.3 изображена диаграмма классов

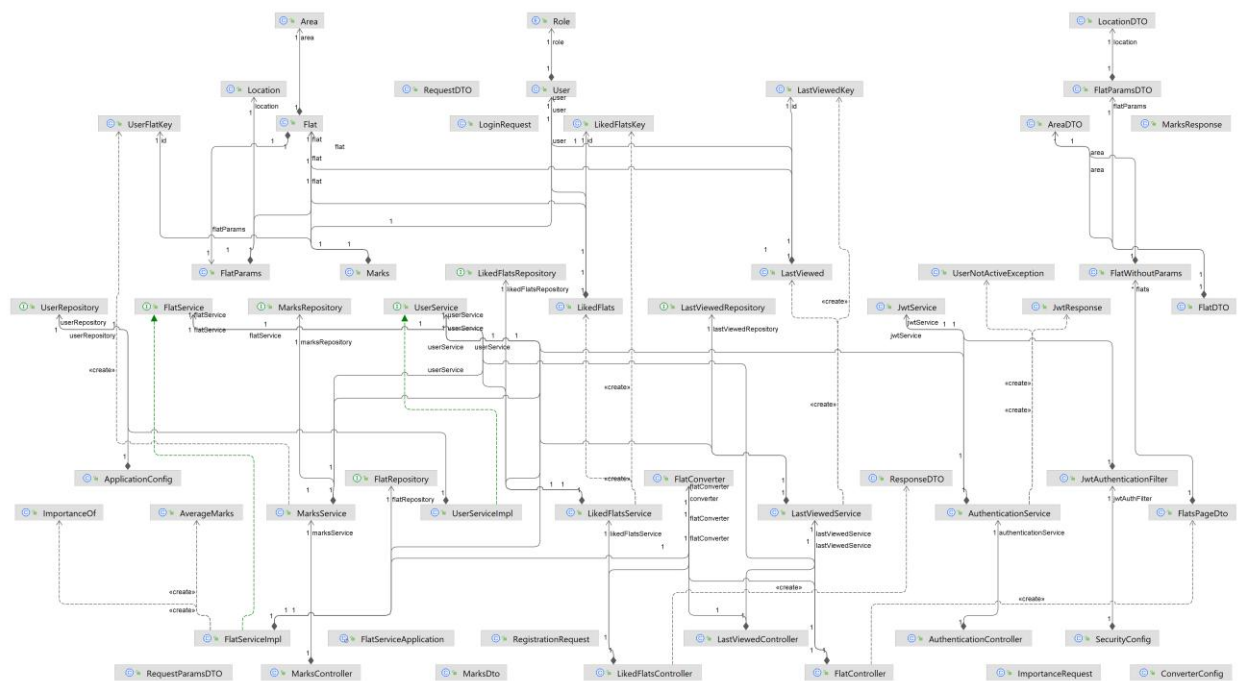


Рисунок 3.3 – Диграмма классов

На рисунке выше изображены классы приложения и их взаимодействие друг с другом. Логика приложения разбита на пакеты, каждый из которых отвечает за отдельные куски логики приложения. Приложение состоит из пакета аутентификации, в котором собрана вся логика по регистрации и авторизации пользователей, из пакета с квартирами, в котором собрана логика по работе с квартирами, а также оценки и пользователи. В каждом из пакетов собраны контроллеры, модели, сервисные слои, а также репозитории, отвечающие за работу с данными.

**Диаграмма состояний.** В основе диаграммы состояний лежат следующие ключевые элементы:

1 Состояния (States). Различные условия, в которых может находиться система или объект. Состояния представлены овалами и часто названы ключевыми словами, описывающими их.

2 События (Events). Изменения, которые могут произойти и привести к переходу между состояниями. События обозначаются стрелками, указывающими направление перехода.

3 Переходы (Transitions). Изменение состояния, которое происходит в ответ на событие. Переходы представлены стрелками, соединяющими состояния и указывающими на событие, вызывающее переход.

4 Начальное состояние (Initial State). Начальное состояние системы перед началом взаимодействия. Обозначается стрелкой, входящей в овал с ключевым словом начального состояния.

5 Конечное состояние (Final State). Состояние, которое обозначает завершение процесса или работы системы. Обозначается овалом с вложенным кругом.

Диаграмма состояний показывает, как система реагирует на различные события и как она изменяет свое состояние в зависимости от текущего контекста. Она широко используется для моделирования жизненного цикла объектов или системы, а также для представления алгоритмов и процессов с различными ветвлениями и условиями.

На рисунке 3.4 изображена диаграмма состояний приложения.

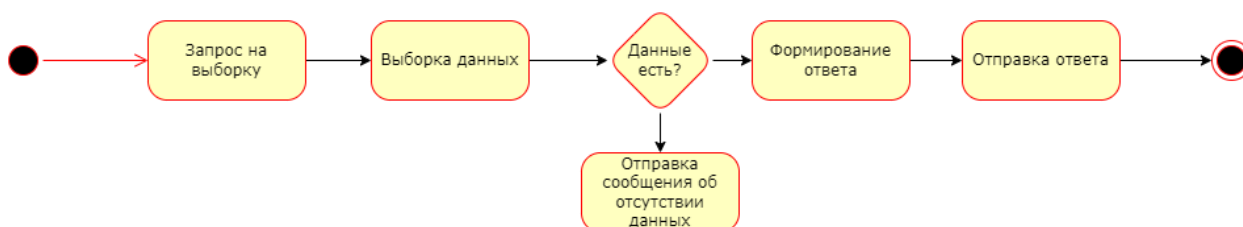


Рисунок 3.4 – Диаграмма состояний

**Диаграмма последовательности.** Диаграмма последовательности (Sequence Diagram) в UML (Unified Modeling Language) представляет собой графическое изображение взаимодействия между объектами или участниками в системе в определенном порядке времени. Эта диаграмма используется для моделирования взаимодействия между различными элементами системы, позволяя визуализировать последовательность сообщений, передаваемых между объектами или участниками в определенном сценарии.

Последовательность действий происходящих в приложении при отправке запроса на просмотр оценённых квартир представлена на рисунке 3.5.

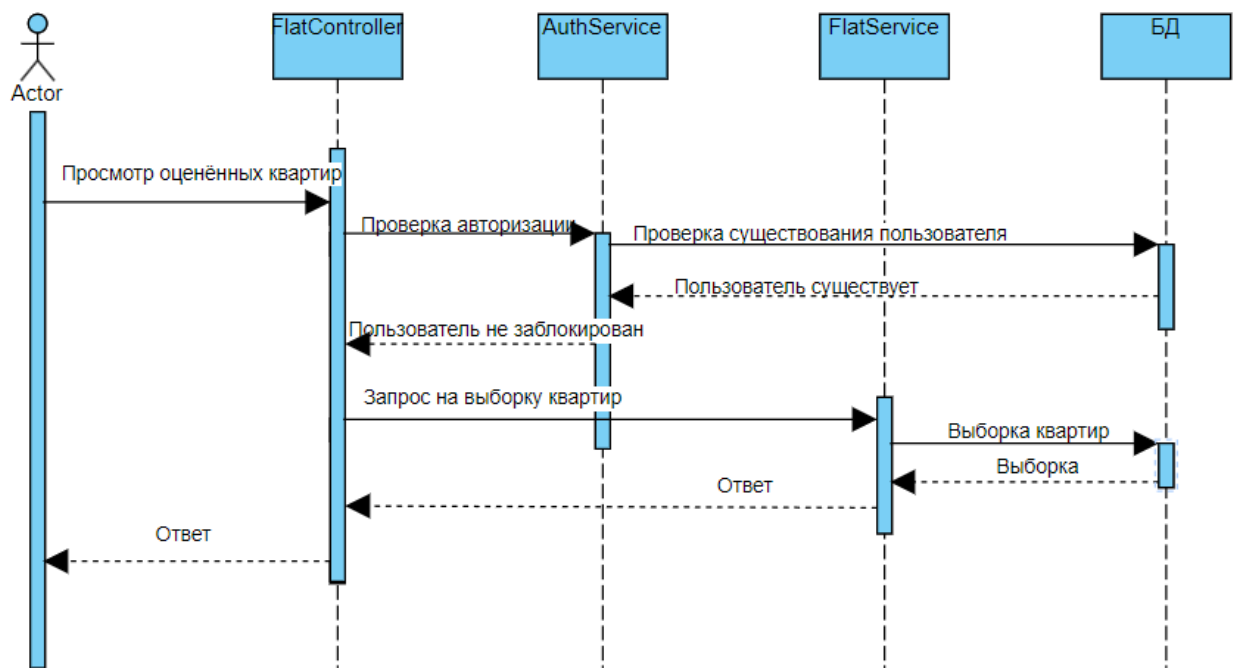


Рисунок 3.5 – Диаграмма последовательности

В начале пользователь отправляет запрос, его обрабатывает контроллер, дальше контроллер передаёт запрос сервису аутентификации для проверки существования пользователя и то, что он не заблокирован, в случае успешного ответа запрос передаётся сервису с квартирами, тот в свою очередь производит запрос в базу данных и далее ответ возвращается пользователю.

### 3.4 Информационная модель системы и её описание

Основным шагом является определение структуры базы данных, которая наилучшим образом отражает характеристики рассматриваемой предметной области. Это включает в себя создание таблиц для хранения данных о квартирах, пользовательских предпочтениях, истории поиска, а также других сущностей, важных для функционирования системы.

Связи между таблицами играют критическую роль в обеспечении целостности данных. Проектирование правильных связей позволяет эффективно извлекать информацию и поддерживать консистентность данных. Например, установление связи между таблицами квартир и пользователя позволяет хранить предпочтения пользователей в контексте конкретных квартир.

Определение индексов и ограничений необходимо для улучшения производительности запросов и обеспечения целостности данных. Индексы

могут быть применены к полям, которые часто используются в запросах, ускоряя процессы поиска и фильтрации данных.

Безопасность данных – важный аспект проектирования базы данных. Это включает в себя установку прав доступа к таблицам, предотвращение несанкционированного доступа к чувствительным данным и шифрование конфиденциальных информации.

Проектирование базы данных – это сложный, но фундаментальный этап в создании программного обеспечения. Эффективная структура базы данных обеспечивает надежность, производительность и легкость сопровождения системы, что является ключевым фактором в обеспечении успешного функционирования автоматизированной системы поддержки принятия решений о приобретении квартиры.

Процесс разработки даталогической модели включает в себя создание таблиц для каждой сущности, определение связей с использованием ключей, установку ограничений целостности, таких как уникальность или необходимость заполнения определенных полей.

На рисунке 3.6 представлена даталогическая модель базы данных.



Рисунок 3.6 – Даталогическая модель базы данных

Данная база данных приведена к 3 нормальной форме [19]. Отношение находится в 3НФ, когда находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа. Проще говоря, второе правило требует выносить все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы в отдельные таблицы.

В данной базе данных представлены следующие сущности:

- пользователи (users);
- квартиры (flats);
- параметры квартиры (flat\_params);
- оценки (marks);
- понравившиеся квартиры (liked\_flats);
- последние просмотренные квартиры (last\_viewed).

Сущность «Пользователи» содержит информацию о пользователях и имеет такие поля как: идентификатор пользователя, его имя, пароль, электронную почту и булево значение активности пользователя.

Сущность «Квартиры» содержит базовую информацию о квартире и имеет такие поля как: идентификатор квартиры, этаж, количество этажей, общую площадь, жилую площадь, площадь кухни, количество комнат, цену, количество просмотров, год постройки, фотографию, дату создания и дату последнего редактирования.

Сущность «Параметры квартиры» содержит дополнительную информацию о квартире и определяется такими полями как: идентификатор квартиры, время до школы, время до детского сада, время до поликлиники, время до метро, время до торгового центра, тип стен, долгота, широта, район, описание квартиры, адрес, булево значение вторичной квартиры и наличие балкона.

Сущность «Оценки» хранит информацию об оценках квартир и идентифицируется пользователем, который поставил оценку, и квартирой, которой была поставлена оценка, присутствует время создания оценки, а также оценки по следующим полям: адрес, район, этаж, общая площадь, площадь кухни, жилая площадь, количество этажей, количество комнат, цена, время до метро, детского сада, школы и поликлиники и торгового центра, год постройки, тип стен

Сущности «Понравившиеся квартиры» и «Последние просмотренные квартиры» одинаковы по своему содержимому и различаются только семантически. В данных сущностях содержатся идентификаторы пользователя и квартиры, а также время сохранения.

### **3.5 Основные алгоритмы программы**

На рисунке Б.1 изображена схема работы уникального поиска.

Сначала пользователь для каждого предоставленного параметра вводит его важность для него, затем исходя из этих оценок важности высчитывается коэффициент важности каждого параметра путём деления единицы на сумму оценок и умножение полученного результата на оценку. После этого высчитывается средние оценки, которые поставили пользователи и сохранены в базе данных. Затем идёт перемножение коэффициента важности на среднее значение оценки и тем самым получаются интегральные показатели предпочтения. После того как интегральный показатель был рассчитан для



каждой квартиры происходит их сортировка от большего к меньшему и результаты выводятся пользователю на экран.

На рисунке Б.2 изображена схема работы просмотра всех квартир.

Первым шагом в данном алгоритме является отправка запроса, который может включать различные, заранее выбранные, фильтры и сортировки, а также номер страницы. Если же фильтры, сортировки и номер выбраны, то они применяются, в ином же случае фильтры не применяются, данные сортируются по умолчанию, и показывается первая страница соответственно. Затем происходит выборка на основе вышеупомянутых критериев и если данные по такому запросу присутствуют, то они отправляются на клиент и происходит показ результата. В противном случае, если данные отсутствуют, происходит отправка сообщения об ошибке и также происходит отображение данного результата.

### 3.6 Руководство пользователя

При старте приложение открывается главная страница (рисунок 3.7). На ней можно увидеть список квартир, различные параметры фильтров, сортировок, а также ссылок на страницы.

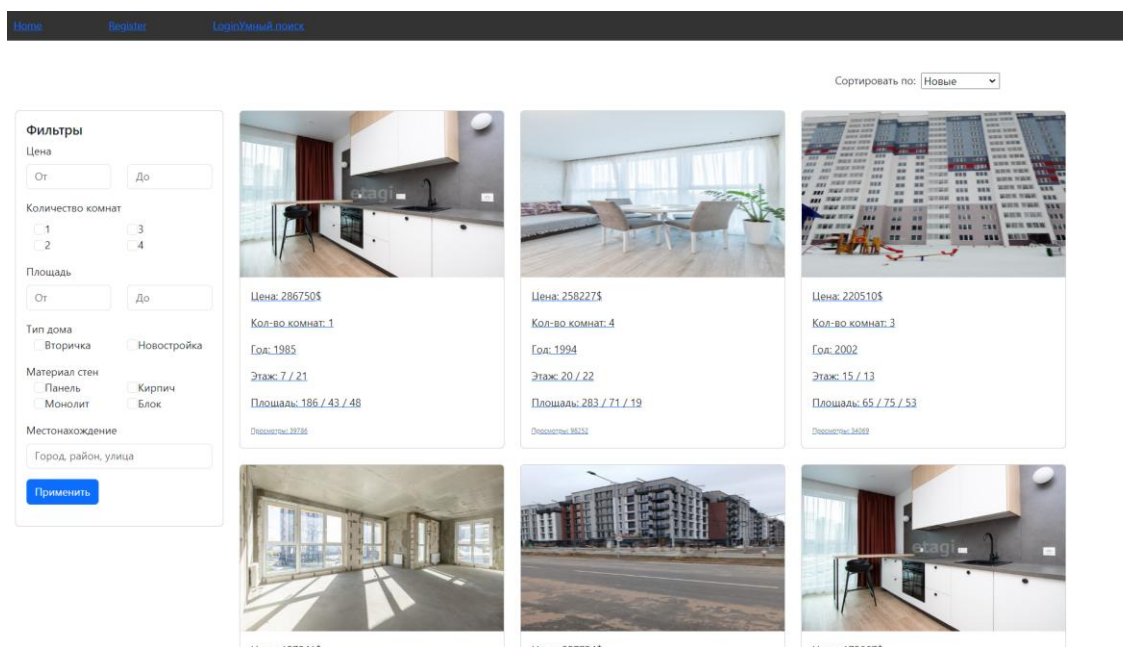


Рисунок 3.7 – Главная страница приложения

Можно применить фильтры (рисунок 3.8) и выдастся подборка квартир соответствующая выбранным фильтрам.

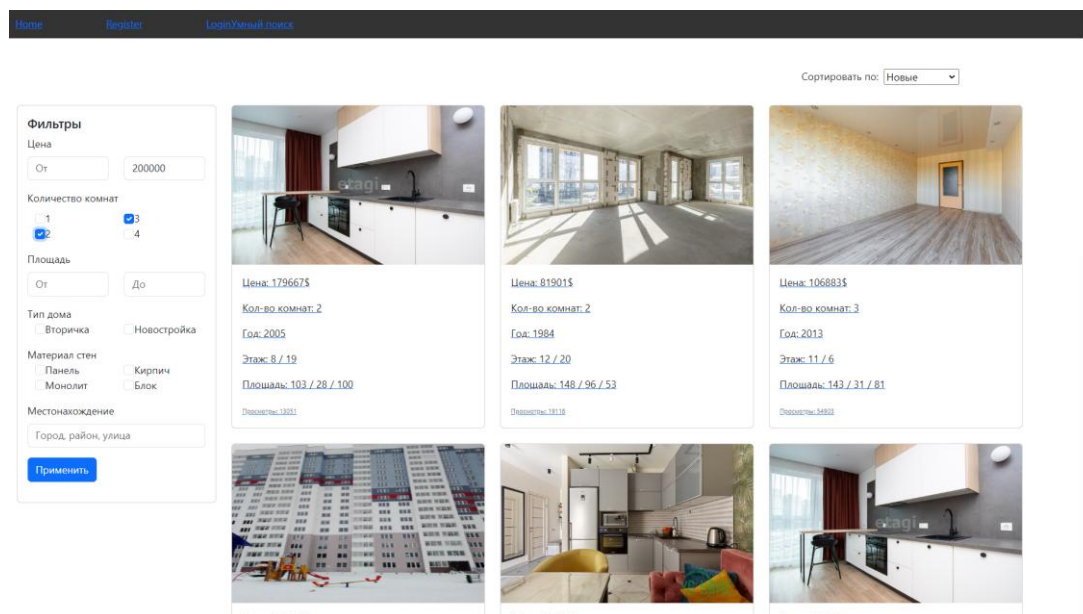


Рисунок 3.8 – Пример работы фильтров

В правом верхнем углу расположена кнопка сортировки, можно применять сортировку, к уже отфильтрованным квартирам (рисунок 3.9).

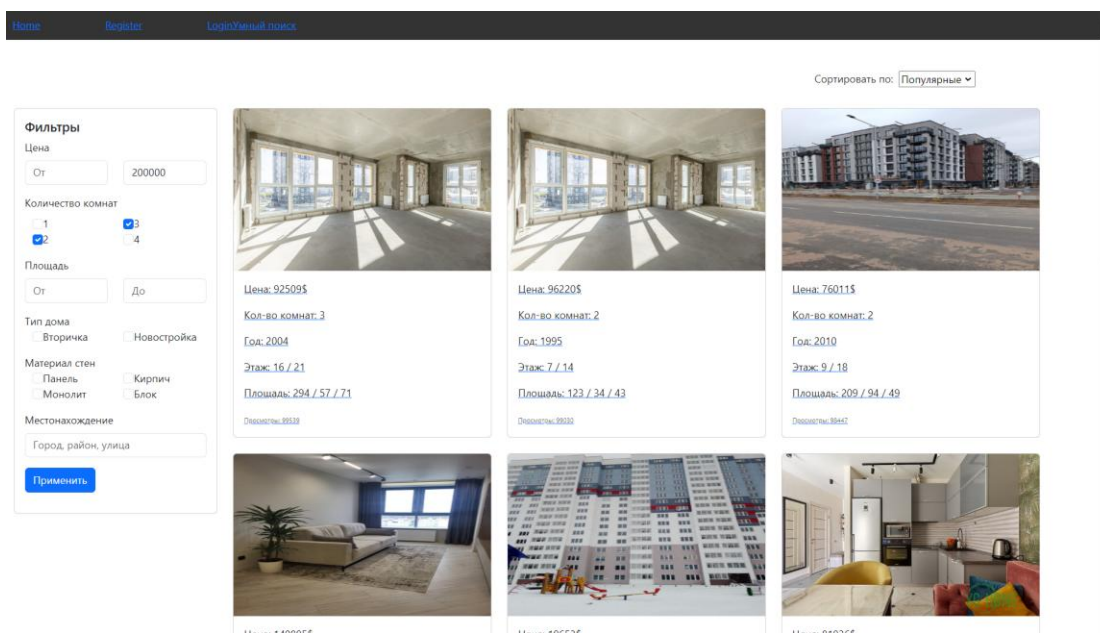


Рисунок 3.9 – Пример работы сортировки с фильтрами

Также если дойти до конца главной страницы, то увидим на ней кнопки пагинации (рисунок 3.10).

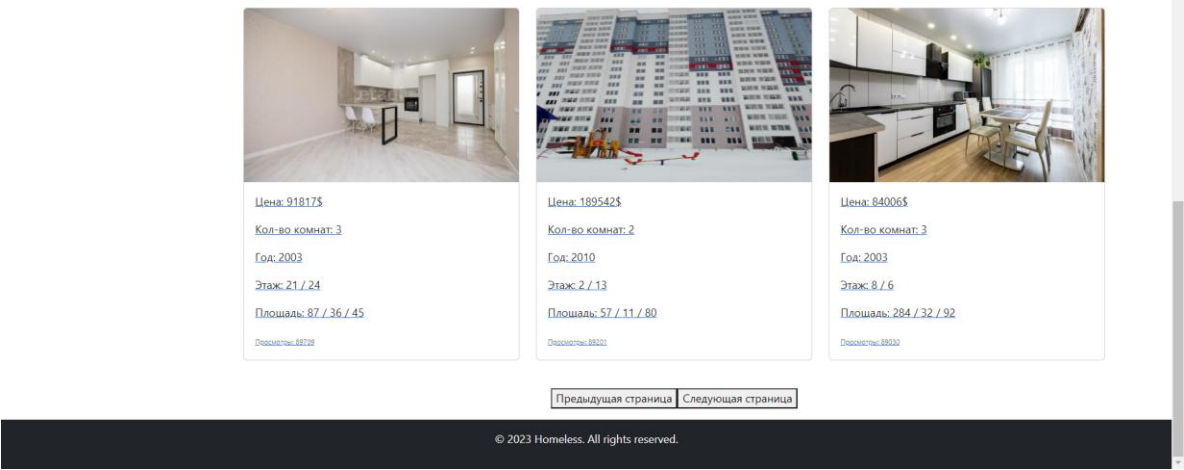


Рисунок 3.10 – Кнопки пагинации

При нажатии на одну из квартир открывается её более детальное описание (рисунок 3.11). На данной странице отображена полная информации о данной квартире со всеми соответствующими параметрами.

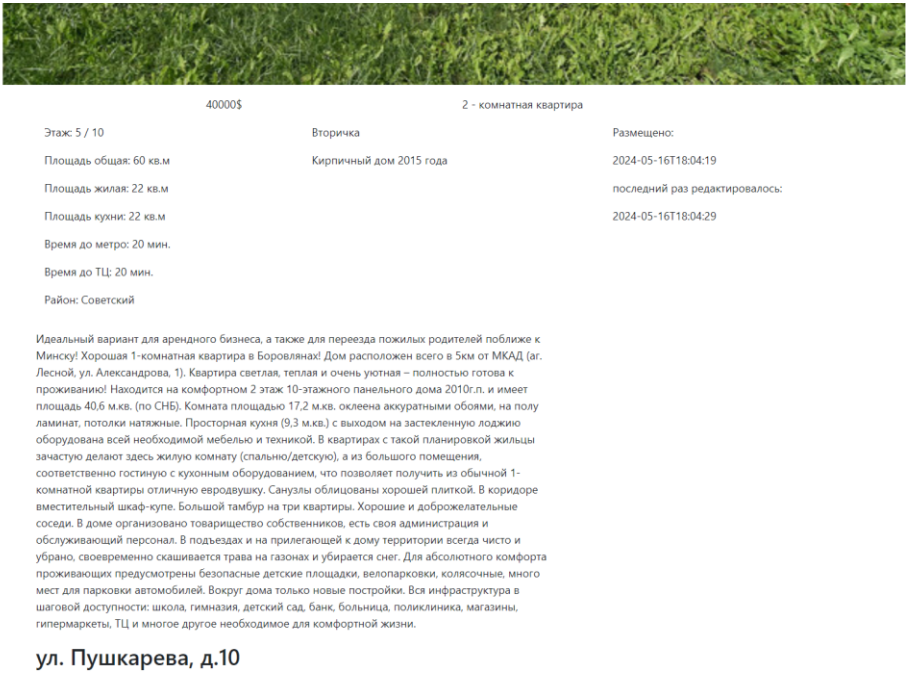


Рисунок 3.11 – Просмотр одной квартиры

При переходе во вкладку «Умный поиск» открывается страница с предпочтениями (рисунок 3.12). На ней предоставляется возможность выбора важности каждого параметра с целью поиска подходящей квартиры.

[Home](#)[Register](#)[LoginУмный поиск](#)

### Умный поиск

Параметр	Важность
Цена	0 ▾
Количество комнат	0 ▾
Год	0 ▾
Этаж	0 ▾
Количество этажей	0 ▾
Общая площадь	0 ▾
Жилая площадь	0 ▾
Площадь кухни	0 ▾
Тип стен	0 ▾
Время до метро	0 ▾
Время до ТЦ	0 ▾
Время до поликлиники	0 ▾
Время до детского сада	0 ▾
Время до школы	0 ▾
Район	0 ▾
Адрес	0 ▾

Найти

Рисунок 3.12 – Вкладка «Умный поиск»

При вводе важности критериев (рисунок 3.13), которые варьируются в значениях от нуля до четырёх, где ноль – неважно, а четыре – очень важно, и нажатии на кнопку «найти» предоставляется уникальная выборка квартир, основанная на важности каждого параметра (рисунок 3.14), более детальный разбор работы данного метода расписан в конце второй главы.

## Умный поиск

Параметр	Важность
Цена	3 ▾
Количество комнат	0 ▾
Год	4 ▾
Этаж	0 ▾
Количество этажей	1 ▾
Общая площадь	4 ▾
Жилая площадь	1 ▾
Площадь кухни	0 ▾
Тип стен	0 ▾
Время до метро	4 ▾
Время до ТЦ	4 ▾
Время до поликлиники	0 ▾
Время до детского сада	0 ▾
Время до школы	0 ▾
Район	3 ▾
Адрес	0 ▾

[Найти](#)

Рисунок 3.13 – Ввод важности

При подборе учитываются все выбранные параметры и их оценки.

При работе приложения, пользователю не нужно знать как происходят расчёты, т.к. ему важен только результат. Весь расчёт происходит на серверной стороне приложения, где происходит расчёт интегральных показателей предпочтения для каждой квартиры и возвращаются результаты от самого подходящего, к наименее подходящему.



## Квартиры на основе ваших предпочтений







 <p><a href="#">Цена: 203996\$</a> <a href="#">Кол-во комнат: 4</a> <a href="#">Год: 2002</a> <a href="#">Этаж: 6 / 9</a> <a href="#">Площадь: 72 / 41 / 7</a> <a href="#">Просмотры: 4931</a></p>	 <p><a href="#">Цена: 73046\$</a> <a href="#">Кол-во комнат: 3</a> <a href="#">Год: 1994</a> <a href="#">Этаж: 20 / 15</a> <a href="#">Площадь: 228 / 7 / 29</a> <a href="#">Просмотры: 82000</a></p>	 <p><a href="#">Цена: 212273\$</a> <a href="#">Кол-во комнат: 1</a> <a href="#">Год: 2007</a> <a href="#">Этаж: 19 / 8</a> <a href="#">Площадь: 44 / 64 / 78</a> <a href="#">Просмотры: 37812</a></p>
 <p><a href="#">Цена: 151686\$</a> <a href="#">Кол-во комнат: 3</a> <a href="#">Год: 2007</a></p>	 <p><a href="#">Цена: 100000\$</a> <a href="#">Кол-во комнат: 4</a> <a href="#">Год: 2020</a></p>	 <p><a href="#">Цена: 84006\$</a> <a href="#">Кол-во комнат: 3</a> <a href="#">Год: 2003</a></p>

Рисунок 3.14 – Уникальная выборка

При переходе во вкладку регистрации открывается форма регистрации (рисунок 3.15). На ней пользователь вводит свои данные в соответствующие поля формы и в случае неправильного ввода, выдаётся сообщение об ошибке.

The screenshot shows a web application header with links: Home, Register, and Login/Умный поиск. Below the header is a 'Register' form. The form has three input fields: 'Name' with the value 'Антон', 'Email' with the value 'anton', and 'Password' with masked characters '.....'. A blue 'Register' button is at the bottom left. A tooltip error message is displayed over the email field, stating: 'Please include an '@' in the email address. 'anton' is missing an '@'.' The tooltip has an orange warning icon.

Рисунок 3.15 – Регистрация

В случае правильного ввода и успешной регистрации выводится сообщение об успехе (рисунок 3.16).

The screenshot shows the same 'Register' form as in Figure 3.15, but with updated values: 'Name' is 'Антон', 'Email' is 'anton@mail.ru', and 'Password' is '.....'. A dark grey success message box is overlaid on the top right of the form, containing the text 'localhost:3000 says Register successful' and an 'OK' button. The blue 'Register' button remains at the bottom left.

Рисунок 3.16 – Успешная регистрация

После регистрации пользователя перенаправляет на главную страницу приложения, где ему становится доступна кнопка личного кабинета, при нажатии на которую он попадает в личный кабинет (рисунок 3.17).

В личном кабинете ему доступен просмотр последних просмотренных квартир, просмотр понравившихся квартир, а также квартир, которые он оценил.

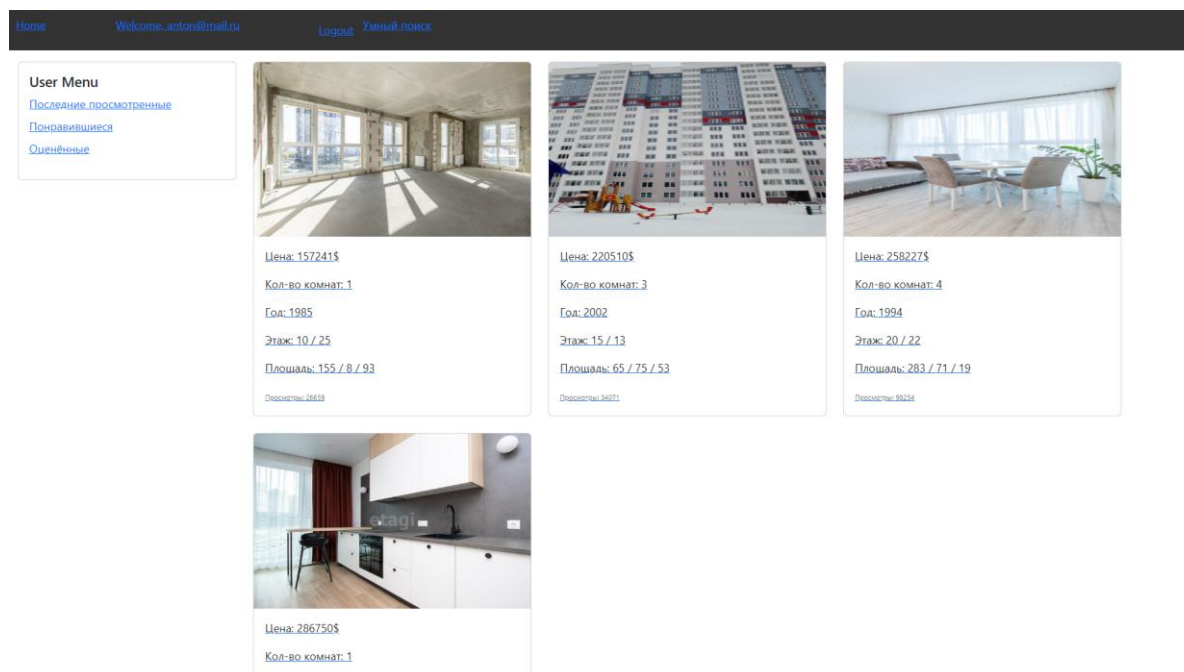


Рисунок 3.17 – Личный кабинет

Если пользователь авторизован и переходит на страницу квартиры ему становится доступны кнопка «лайк» и кнопка оценить квартиру(рисунок 3.18).

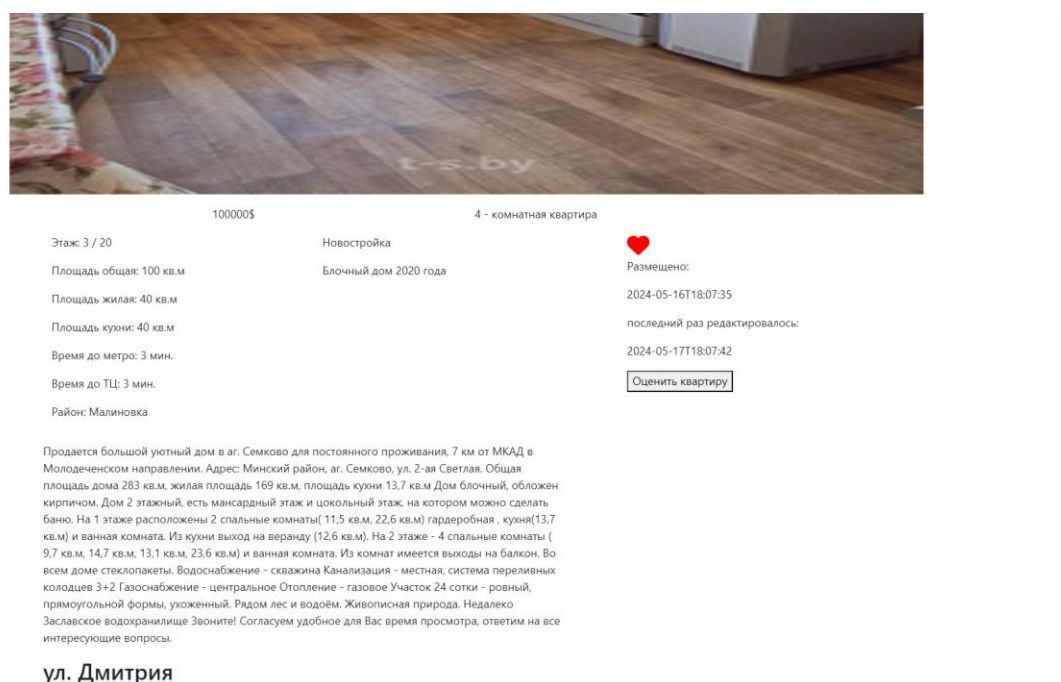


Рисунок 3.18 – Кнопки



При нажатии на кнопку «лайк» она загорается красным и добавляется в список понравившихся квартир. При нажатии на кнопку «Оценить квартиру» пользователь попадает на страницу оценки квартиры (рисунок 3.19).

Параметр	Значение	Оценка
Цена	100000\$	1 ▼
Количество комнат	4	1 ▼
Год	2020	1 ▼
Этаж	3	1 ▼
Количество этажей	20	1 ▼
Общая площадь	100	1 ▼
Жилая площадь	40	1 ▼
Площадь кухни	40	1 ▼
Материал стен	BLOCK	1 ▼
Время до метро	3	1 ▼
Время до ТЦ	3	1 ▼
Время до поликлиники	3	1 ▼
Время до сада	3	1 ▼
Время до школы	3	1 ▼
Район	Малиновка	1 ▼
Адресс	ул. Дмитрия	1 ▼

Оценить

Рисунок 3.19 – Страница оценки квартиры

Здесь пользователь может поставить оценку каждому параметру, основываясь на значениях каждого параметра. После оценки он может посмотреть оценённые квартиры в личном кабинете (рисунок 3.20).

Home Welcome: anton@mail.ru Logout Умный поиск

User Menu

- Последние просмотренные
- Понравившиеся
- Оценённые

To exit full screen, press and hold Esc

Цена: 100000\$

Кол-во комнат: 4

Год: 2020

Этаж: 3 / 20

Площадь: 100 / 40 / 40

Просмотреть: 1000\$

Рисунок 3.20 – Просмотр оценённых квартир

Также если пользователь нажал на кнопку «лайк» на страницы какой-либо квартиры, то данная квартира появится в его личном кабинете во вкладке «понравившиеся» (рисунок 3.21).

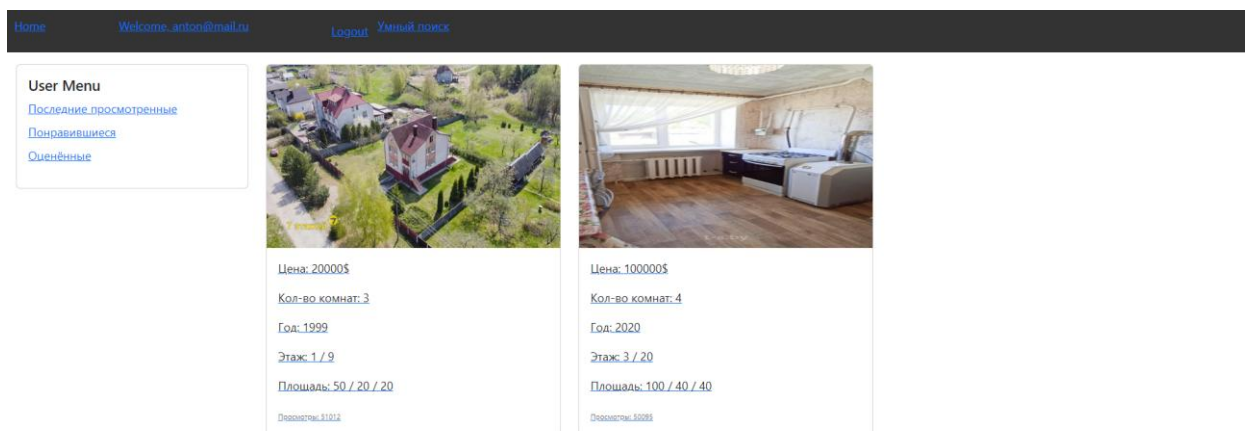


Рисунок 3.21 – Просмотр понравившихся квартир

Также можно покинуть систему и тогда пользователь будет перенаправлен на главную страницу без личного кабинета. Чтобы обратно войти в систему, пользователь может ввести свои данные на странице авторизации (рисунок 3.22).



Рисунок 3.22 – Страница авторизации

Таким образом и была разработана программная реализация системы поддержки принятия решений о приобретении квартиры.

## **4 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ О ПРИОБРЕТЕНИИ КВАРТИРЫ**

### **4.1 Характеристика программного продукта**

Разработанное программное средство представляет собой веб-приложение для помощи поиска квартир.

Программное обеспечение позволяет:

- фильтровать и сортировать квартиры;
- отображать информацию о квартире;
- уведомлять об изменении цены;
- просматривать список квартир.

Разработка программного средства осуществлялась на базе платформы Spring.

Разработанное программное обеспечение предназначено для пользователей сети Интернет.

Разрабатываемое программное средство имеет следующие преимущества по сравнению с существующими аналогами:

- удобный и информативный интерфейс;
- простота использования;
- возможность предоставлять персонализированную выборку

квартир.

Реализация программного средства на рынке планируется через размещение веб-приложения в сети интернет версии, в которой будет присутствовать реклама

### **4.2 Расчёт затрат на разработку программного продукта**

Общий объем программного модуля определяется на основе информации о функциях разрабатываемого программного модуля, исходя из количества и объема функций, реализуемых программным модулем, по формуле:

$$V_o = \sum_{i=1}^n V_i, \quad (4.1)$$

где  $n$  – общее число функций;

$V_i$  – объем  $i$ -ой функции программного модуля.

С учетом условий разработки общий объем программного модуля уточняется в организации и определяется уточненный объем программного модуля по формуле:

$$V_y = \sum_{i=1}^n V_{yi}, \quad (4.2)$$

где  $V_{yi}$  – уточненный объем  $i$ -й функции программного модуля (LOC).

Определение общего объема программного модуля в соответствии с содержанием функций представлено в таблице 4.1.

Таблица 4.1 – Расчет общего объема программного продукта

Номер функции	Наименование (содержание) функции	Объем функции (LOC)	
		по каталогу	уточненный
101	Организация ввода информации	150	150
109	Организация ввода/вывода информации в интерактивном режиме	320	320
203	Формирование баз данных	500	360
204	Обработка наборов и записей базы данных	500	320
208	Организация поиска и поиск в базе данных	300	150
506	Обработка ошибочных и сбойных ситуаций	300	150
707	Графический вывод результатов	2000	1800
	Общий объем программного модуля	4070	3250

В связи с использованием более совершенных средств автоматизации объемы функций были уменьшены и уточненный объем программного модуля составил 4070 LOC вместо 3250 LOC.

Программный модуль относится ко второй категории сложности, следовательно, нормативная трудоемкость составляет 81 чел./дн.

Нормативная трудоемкость служит основой для определения общей трудоемкости разработки программного модуля, который определяется по формуле:

$$T_o = T_n \cdot K_c \cdot K_T \cdot K_H, \quad (4.3)$$

где  $T_n$  – нормативная трудоемкость разработки программного модуля;  
 $K_c$  – коэффициент, учитывающий сложность программного модуля.

Коэффициент, учитывающий сложность программного модуля рассчитывается по формуле:

$$K_c = 1 + \sum_{i=1}^n K_i, \quad (4.4)$$

где  $n$  – количество учитываемых характеристик;

$K_i$  – коэффициент, соответствующий степени повышения сложности программного модуля за счет  $i$ -ой характеристики;

$K_T$  – поправочный коэффициент, учитывающий степень использования при разработке стандартных модулей;

$K_H$  – коэффициент, учитывающий степень новизны программного модуля.

Коэффициент сложности составит:

$$K_c = 1 + 0,12 = 1,12.$$

Коэффициенты использования стандартных модулей и новизны программного модуля составят  $K_T = 0,6$  и  $K_H = 0,8$  соответственно.

Таким образом, общая трудоемкость разработки программного модуля составит:

$$T_o = 81 \cdot 1,12 \cdot 0,6 \cdot 0,8 = 44 \text{ чел./дн.}$$

В соответствии с договором срок разработки – 1,5 месяца.

Численность исполнителей проекта ( $Ч_p$ ) рассчитывается по формуле:

$$Ч_p = \frac{T_o}{T_p \cdot \Phi_{эф}}, \quad (4.5)$$

где  $\Phi_{эф}$  – годовой эффективный фонд времени работы одного работника;

$T_o$  – общая трудоемкость разработки проекта, чел./дн.;

$T_p$  – срок разработки проекта, лет.

Эффективный фонд времени работы одного работника ( $\Phi_{эф}$ ) рассчитывается по формуле:

$$\Phi_{\text{эф}} = D_{\text{г}} - D_{\text{п}} - D_{\text{в}} - D_{\text{о}}, \quad (4.6)$$

где  $D_{\text{г}}$  – количество дней в году;

$D_{\text{п}}$  – количество праздничных дней в году;

$D_{\text{в}}$  – количество выходных дней в году;

$D_{\text{о}}$  – количество дней отпуска.

По формуле 4.6 эффективный фонд времени составит:

$$\Phi_{\text{эф}} = 366 - 10 - 103 - 24 = 229 \text{ дн.}$$

Численность разработчиков программного модуля составит:

$$Ч_{\text{р}} = \frac{44}{0,125 \cdot 229} = 2 \text{ чел.}$$

Для разработки программного средства понадобятся: Middle программист и Senior программист.

Расчет основной заработной платы исполнителей представлен в таблице 4.2.

Таблица 4.2 – Расчет основной заработной платы исполнителей

Категория исполнителей	Эффективный фонд времени работы, дн.	Дневная тарифная ставка, руб.	Тарифная заработная плата, руб.
Middle программист	14	100	1400
Senior программист	30	200	6000
Всего	44		7400
Премия (50 %)			3700
Основная заработная плата			11100

Дополнительная заработная плата исполнителей, занятых разработкой программного модуля, рассчитывается по формуле:

$$З_{\text{д}} = \frac{З_{\text{о}} \cdot Н_{\text{д}}}{100}, \quad (4.7)$$

где  $Н_{\text{д}}$  – норматив дополнительной заработной платы (10%).

Дополнительная заработная плата исполнителей составит:

$$З_{\text{д}} = \frac{11100 \cdot 10}{100} = 1100 \text{ руб.}$$

Отчисления в фонд социальной защиты населения и на обязательное страхование рассчитываются в соответствии с действующим законодательством по формуле:

$$З_{\text{сз}} = \frac{(З_{\text{o}} + З_{\text{д}}) \cdot Н_{\text{сз}}}{100}, \quad (4.8)$$

где  $Н_{\text{сз}}$  – норматив отчислений в фонд социальной защиты населения и на обязательное страхование (35%).

Отчисления в ФСЗН и на обязательное страхование составят:

$$З_{\text{сз}} = \frac{(11100 + 1100) \cdot 35}{100} = 4273,5 \text{ руб.}$$

Затраты по статье «Материалы» определяется по формуле:

$$М = Н_{\text{м}} \frac{V_{\text{o}}}{100}, \quad (4.9)$$

где  $Н_{\text{м}}$  – норма расхода материалов в расчете на 100 строк исходного кода программного модуля (0,38 руб.);

$V_{\text{o}}$  – общий объем программного модуля (3250 LOC).

Таким образом, затраты на материалы составят:

$$М = 0,38 \cdot \frac{3250}{100} = 12,35 \text{ руб.}$$

Затраты по статье «Машинное время» рассчитываются по формуле:

$$Р_{\text{м}} = Ц_{\text{м}} \frac{V_{\text{o}}}{100} \cdot Н_{\text{мв}}, \quad (4.10)$$

где  $Ц_{\text{м}}$  – цена одного машино-часа (1 руб.);

$Н_{\text{мв}}$  – норматив расхода машинного времени на отладку 100 строк исходного кода (12 машино-часов).

Таким образом, затраты на машинное время составят:

$$P_m = 1 \cdot 32,5 \cdot 12 = 390 \text{ руб.}$$

Накладные расходы рассчитываются по формуле:

$$P_n = \frac{3_o \cdot H_n}{100}, \quad (4.11)$$

где  $H_n$  – норматив накладных расходов (100 %).

Нормативные расходы составят:

$$P_n = \frac{11100 \cdot 100}{100} = 11100 \text{ руб.}$$

Общая сумма затрат на разработку программного средства составит:

$$C_p = 11100 + 1100 + 4273,5 + 12,35 + 390 + 11100 = 27975,85 \text{ руб.}$$

### **4.3 Оценка экономического эффекта от реализации программного продукта**

Целевой аудиторией разрабатываемого программного средства являются пользователи сети интернет.

Так как невозможно предугадать количество пользователей, которые будут посещать сайт и в соответствии с этим устанавливать цену. Поэтому исходя из данных Realt о ценах на размещению рекламы [20] возьмём примерную стоимость рекламы в месяц равную 3000 руб. И в зависимости от трафика сайта можем регулировать количество показов. Прогнозный доход от размещения платной рекламы в веб-приложении составит 36000 руб. в год. Чистая прибыль, которую получит организация от реализации программного средства, определяется по формуле:

$$П_ч = (D_p - \text{НДС}) \cdot (1 - H_n), \quad (4.12)$$

где  $D_p$  – доход от размещения рекламы в веб-приложении, руб;

НДС – сумма налога на добавочную стоимость, руб;

$H_n$  – сумма налога на прибыль, в долях единицы (0,20).



Налог на добавленную стоимость с доходов от размещения рекламы определяется по формуле:

$$\text{НДС} = \frac{D_p \cdot H_{\text{дс}}}{100\%}, \quad (4.13)$$

где  $H_{\text{дс}}$  – ставка налога на добавленную стоимость (20%).

Сумма налога на добавленную стоимость составит:

$$\text{НДС} = \frac{36000 \cdot 20}{100} = 7200 \text{ руб.}$$

Чистая прибыль от реализации программного продукта в первый год составит:

$$П_q = (36000 - 7200)(1 - 0,20) = 23040 \text{ руб.}$$

#### **4.4 Расчёт показателей эффективности инвестиций в разработку ПО**

В процессе использования программного средства чистая прибыль в конечном итоге возмещает капитальные затраты. Однако, полученные при этом суммы результатов (прибыли) и затрат (капитальных вложений) по годам приводят к единому времени – расчетному году (за расчетный год принят 2024 год) путем умножения результатов и затрат за каждый год на коэффициент дисконтирования, который рассчитывается по формуле:

$$a_t = \frac{1}{(1 + E_n)^{t - t_p}}, \quad (4.14)$$

где  $E_n$  – норматив приведения разновременных затрат и результатов;

$t_p$  – расчетный год;

$t$  – номер года, который приводится к расчётному.

Норматив приведения разновременных затрат и результатов рассчитывается по формуле:

$$E = r + s + \sum_{i=1}^n gi, \quad (4.15)$$

где  $r$  – реальная (без учета компенсации за инфляцию) безрисковая ставка  
ссудного процента;  
 $s$  – инфляционное ожидание за период  $t$ , рассчитанное как среднее за  
расчетный период проекта;  
 $i$  – факторы риска;  
 $g$  – премия за конкретный риск по отдельному фактору.

В расчет возьмем процентную ставку по депозитам в белорусских  
рублях равную 11,5%. Уровень инфляции по данным НБРБ за год составил 8%.  
Премию за конкретный риск по отдельному фактору  $g$  примем также 1%.

Таким образом норматив приведения разновременных затрат и  
результатов равен:

$$E_n = 0,115 + 0,08 + 0,01 = 0,205.$$

Коэффициентам приведения по годам (2024 – 2027) будут  
соответствовать следующие значения:

$$\begin{aligned}\alpha_1 &= (1 + 0,205)^{1-1} = 1; \\ \alpha_2 &= (1 + 0,205)^{1-2} = 0,83; \\ \alpha_3 &= (1 + 0,205)^{1-3} = 0,69; \\ \alpha_4 &= (1 + 0,205)^{1-4} = 0,57.\end{aligned}$$

Результаты расчета показателей эффективности приведены в таблице  
4.3.

Таблица 4.3 – Показатели эффективности внедрения ПС

Показатель	Ед. изм.	По годам эксплуатации			
		2024	2025	2026	2027
1	2	3	4	5	6
1. Прирост чистой прибыли	руб.	11520	23040	23040	23040
2. То же с учетом фактора времени	руб.		19123,2	15897,6	13132,8
3. Инвестиции в разработку	руб.	27972,85	-	-	-

Продолжение таблицы 4.3

1	2	3	4	5	6
4. То же с учетом фактора времени	руб.	27972,85	-	-	-
5. Чистый дисконтированный доход	руб.	-16452,85	19123,2	15897,6	13132,8
6. Чистый дисконтированный доход с нарастающим итогом	руб.	-16452,85	2670,35	18567,95	31700,75
7. Коэффициент дисконтирования ( $a_t$ )		1	0,83	0,69	0,57

Рентабельность инвестиций в разработку программного модуля составит:

$$P_{\text{и}} = \frac{P_{\text{ч}}}{C_{\text{р}}} \cdot 100\% = \frac{(11520 + 23040 \cdot 3)/4}{27972,85} \cdot 100\% = 72 \%$$

В результате экономическое обоснование программного средства показало, что чистый дисконтированный доход за четыре года эксплуатации программного средства составит 31700,75 руб. Все инвестиции окупаются на второй год.

Таким образом создание автоматизированной системы помощи принятия решений о покупке квартиры является экономически эффективным и инвестиции в его разработку целесообразно осуществлять.

## ЗАКЛЮЧЕНИЕ

В результате работы над дипломным проектом была достигнута поставленная цель, решены поставленные задачи. В рамках данного проекта рассматривалась проблематика выбора квартиры, которая является актуальной и имеет место для дальнейшего развития.

Было проведено исследование процесса поиска и покупки недвижимости, и было выявлено, что рынок недвижимости велик и не стагнирует, а наоборот пользуется значительным спросом.

Также были проанализированы бизнес-процессы, связанные с покупкой квартиры, были выявлены существующие проблемы и предложены пути решения в виде автоматизированной системы. В связи с этим для наглядности был продемонстрирован декомпозированный процесс покупки квартиры без применения автоматизации, а также с её применением, а именно с использованием экспертных оценок, при помощи которого этот процесс автоматизируется.

Были поставлены задачи, осуществлено проектирование при помощи UML диаграмм. Осуществлялась разработка программного средства с помощью обоснованных технологий и в итоге было сделано приложение, с описанием его работы, которое помогает пользователям экономить время и силы на поиске и принятии решений. В результате поставленные задачи были выполнены и программное является надёжным помощником в решении задач принятия решений. Данное решение можно улучшить с помощью, набирающего популярность, искусственного интеллекта, обучение которого будет происходить на основании данных полученных от пользователей.

В результате экономического обоснования было выявлено, что чистый дисконтированный доход за четыре года составит 31700,75 руб, а вложенные инвестиции окупятся на второй год использования.

Таким образом создание автоматизированной системы помощи принятия решений о покупке квартиры является экономически эффективным и выгодным, и инвестиции в его разработку целесообразно осуществлять.

Подводя итог выполненной работы, была подтверждена актуальность проекта, поставленные задачи выполнены, требования дипломного проекта учтены.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] C-ens [Электронный ресурс]. – Электронные данные. - Режим доступа: <https://c-ens.by/analytics/kak-kupit-kvartiru-bez-posrednikov-osnovnye-etapy-i-vazhnye-njuansy/>.
- [2] Realty [Электронный ресурс]. – Электронные данные. - Режим доступа: <https://realty.rbc.ru/news/645017d79a79473f67427f86>.
- [3] Устюшенко, Н. А. Рынок недвижимости: Республики Беларусь / Н. А. Устюшенко // Экономика, финансы, управление. – 2012. – № 2 (146). – С. 4–7.
- [4] Устюшенко, Н. А. Рынок недвижимости: этапы развития и настоящее / Н. А. Устюшенко // Экономика, финансы, управление. – 2012. – № 2 (146). – С. 29–41.
- [5] Национальное кадастровое агентство [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://pr.nca.by/>.
- [6] Realt [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://realt.by/news/article/39528/>.
- [7] Belmarket [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://belmarket.by/news/news-56643.html>.
- [8] Habr [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://habr.com/ru/sandbox/161642/>.
- [9] Садовникова, Н. П. Системы поддержки принятия решений : справ. пособие / Н. П. Садовникова, Д. С. Парыгин, М. В. Щербаков. – Волгоград , 2021. – 110 с.
- [10] Петровский, А. Б. Теория принятия решений / А. Б. Петровский. – М. : Академия, 2009. – 400 с.
- [11] Эккель Б. Философия Java. 4-е полное изд. — СПб.: Питер, 2018. — 1168 с.
- [12] Структуры данных и алгоритмы Java / Р. Лафоре. - 2-е изд. - Санкт-Петербург : Питер,. 2018. - 704 с.
- [13] Г. Шилд, Java 8. Руководство для начинающих, 2015.
- [14] React [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://react.dev/>.
- [15] Habr [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://habr.com/ru/articles/38730/>.
- [16] PostgreSQL [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.postgresql.org>.

[17] Gameinstitute [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://gameinstitute.ru/resources/lessons/uml/4-diagrammyi-sostoyaniy-konechnyih-avtomatov/>.

[18] Flexberry [Электронный ресурс]. – Электронные данные. – Режим доступа: [https://flexberry.github.io/ru/gpg\\_class-diagram.html/](https://flexberry.github.io/ru/gpg_class-diagram.html/).

[19] Habr [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://habr.com/ru/articles/254773/>.

[20] Realt [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://realt.by/price-list/>.

## ПРИЛОЖЕНИЕ А

### (обязательное)

### Отчёт о проверке в системе «Антиплагиат»

Оригинальность	98,18%	Совпадения	1,82%	Цитирования	0%	Самоцитирования	0%
----------------	--------	------------	-------	-------------	----	-----------------	----

[ПОЛНЫЙ ОТЧЕТ](#) [КРАТКИЙ ОТЧЕТ](#) [ИСТОРИЯ ОТЧЕТОВ](#) [РАСПЕЧАТАТЬ](#) [ВЫГРУЗИТЬ](#) [СОЗДАТЬ ССЫЛКУ](#)

Свойства документа

Структура документа

Текстовые метрики **NEW**

Авторы документа ?

Имя исходного файла

Название документа

Тип документа

Не указано

Не указано

diplom.pdf

diplom.pdf

Не указано

Рисунок А.1 – Отчёт о проверке в системе антиплагиат

## ПРИЛОЖЕНИЕ Б (обязательное)

### Основные алгоритмы работы программы



Рисунок Б.1 – Схема работы уникального поиска



## Продолжение приложения Б

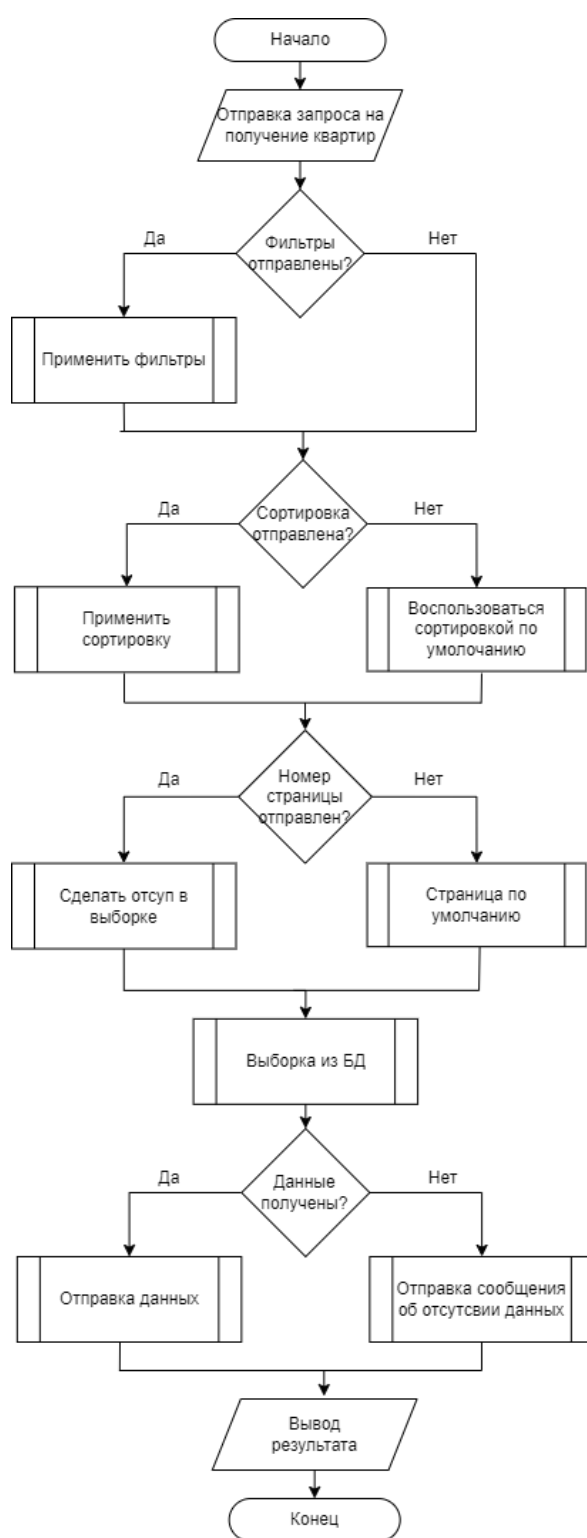


Рисунок Б.2 – Схема работы просмотра всех квартир

## ПРИЛОЖЕНИЕ В

### (обязательное)

#### Листинг скрипта генерации базы данных

```
create table if not exists public.flats
(
    floor            integer,
    kitchen_area     integer,
    living_area      integer,
    number_of_floors integer,
    number_of_rooms  integer,
    number_of_views  integer,
    price            double precision,
    total_area       integer,
    year            integer,
    created_at       timestamp(6),
    id               bigint not null
        primary key,
    last_time_up     timestamp(6),
    photo            varchar(350)
);

alter table public.flats
    owner to username;

create table if not exists public.flat_params
(
    has_balcony      boolean,
    resale            boolean,
    time_to_mall      integer,
    time_to_metro     integer,
    flat_id           bigint not null
        primary key
        constraint fkm3q2c94l4b1k0cvpt11v9a99b
            references public.flats,
    address           varchar(255),
    description       text,
    district          varchar(255),
    latitude          varchar(255),
    longitude         varchar(255),
    walling           varchar(255),
    time_to_clinic    integer,
    time_to_kindergaten integer,
    time_to_school    integer
);

alter table public.flat_params
    owner to username;

create table if not exists public.users
(
    is_active boolean default true,
    id         bigint      not null
        primary key,
    email      varchar(255) not null
        unique,
    name       varchar(255) not null,
```

## Продолжение приложения В

```
password varchar(255) not null,
role      varchar(255)
constraint users_role_check
check ((role)::text = ANY
      ((ARRAY ['USER'::character varying, 'ADMIN'::character
varying, 'EXPERT'::character varying])::text[]))
);

alter table public.users
owner to username;

create table if not exists public.last_viewed
(
flat_id    bigint not null
constraint fk4lhnlgvjsf3vdfbtu8rcirrgs
references public.flats,
timestamp timestamp(6),
user_id    bigint not null
constraint fkjpk0b9j19ejrmb45ryytlvnt
references public.users,
primary key (flat_id, user_id)
);

alter table public.last_viewed
owner to username;

create table if not exists public.liked_flats
(
flat_id    bigint not null
constraint fk5ledcpt6oxffo8bull101p8d6e
references public.flats,
timestamp timestamp(6),
user_id    bigint not null
constraint fkh2taechlgry6nvr8ki5ftr14l
references public.users,
primary key (flat_id, user_id)
);

alter table public.liked_flats
owner to username;

create table if not exists public.marks
(
flat_id    bigint not null
constraint fknqtpxjcenmcpok4vy2ltbueii
references public.flats,
user_id    bigint not null
constraint fkexm2itxq6fj3pll51arpieejd
references public.users,
address     integer,
created_at  timestamp(6),
district    integer,
floor       integer,
kitchen_area integer,
living_area integer,
number_of_floors integer,
number_of_rooms integer,
price       integer,
time_to_mall integer,
```

## Продолжени приложения В

```
time_to_metro      integer,  
total_area         integer,  
walling            integer,  
year               integer,  
time_to_clinic     integer,  
time_to_kindergaten integer,  
time_to_school     integer,  
primary key (flat_id, user_id)  
);  
  
alter table public.marks  
owner to username;
```

## ПРИЛОЖЕНИЕ Г (обязательное)

### Листинг кода

```
@Service
@RequiredArgsConstructor
public class FlatServiceImpl implements FlatService {

    private final FlatRepository flatRepository;
    private final EntityManager em;
    private final FlatConverter converter;
    private final UserService userService;
    private static final int DEFAULT_PAGE_SIZE = 12;

    @Override
    public List<Flat> getAll() {
        return flatRepository.findAll();
    }

    public Flat getOne(long id) {
        Flat flat =
flatRepository.findById(id).orElseThrow(NoSuchElementException::new);

        return flat;
    }

    public Page<Flat> getByCriteria(RequestParamsDTO params) {
        CriteriaBuilder builder = em.getCriteriaBuilder();
        CriteriaQuery<Flat> query = builder.createQuery(Flat.class);
        Root<Flat> root = query.from(Flat.class);
        Join<Flat, FlatParams> rootFlatParams = root.join("flatParams");
        List<Predicate> predicates = new ArrayList<>();

        if (params.getMinPrice() != null) {
            predicates.add(builder.greaterThanOrEqualTo(root.get("price"),
params.getMinPrice()));
        }
        if (params.getMaxPrice() != null) {
            predicates.add(builder.lessThanOrEqualTo(root.get("price"),
params.getMaxPrice()));
        }
        if (params.getNumberOfRooms() != null &&
!params.getNumberOfRooms().isEmpty()) {
predicates.add(root.get("numberOfRooms").in(params.getNumberOfRooms()));
        }
        if (params.getMinSquare() != null) {
predicates.add(builder.greaterThanOrEqualTo(root.get("area").get("total"),
params.getMinSquare()));
        }
        if (params.getMaxSquare() != null) {
predicates.add(builder.lessThanOrEqualTo(root.get("area").get("total"),
params.getMaxSquare()));
        }
        if (params.getResale() != null && !params.getResale().isEmpty()) {
```

## Продолжение приложения Г

```
predicates.add(rootFlatParams.get("resale").in(params.getResale()));
    }

    if (params.getWalling() != null && !params.getWalling().isEmpty()) {
predicates.add(rootFlatParams.get("walling").in(params.getWalling()));
    }
    if (params.getAddress() != null && !params.getAddress().isEmpty()) {
predicates.add(builder.like(rootFlatParams.get("location").get("address"),
"% " + params.getAddress() + "%"));
    }

    query.where(
        builder.and(predicates.toArray(new Predicate[0]))
    );

    if (params.getSortBy() == null) {
        query.orderBy(builder.desc(root.get("lastTimeUp")));
    } else {
        String[] split = params.getSortBy().split(":");
        String direction = split[1];
        String sortParam = split[0];
        if (direction.equals("asc")) {
            query.orderBy(builder.asc(root.get(sortParam)));
        } else {
            query.orderBy(builder.desc(root.get(sortParam)));
        }
    }
}

TypedQuery<Flat> typedQuery = em.createQuery(query);

int page = params.getPage();
typedQuery.setFirstResult(page * DEFAULT_PAGE_SIZE);
typedQuery.setMaxResults(DEFAULT_PAGE_SIZE);

List<Flat> resultList = typedQuery.getResultList();

if (resultList.isEmpty()) {
    throw new NoResultException("Нет квартир по такому запросу");
}

Long totalRecords = getTotalRecords(em, params);
int totalPages = (int) Math.ceil((double) totalRecords /
DEFAULT_PAGE_SIZE);

return new PageImpl<>(resultList, PageRequest.of(page,
DEFAULT_PAGE_SIZE), totalPages);
}

private Long getTotalRecords(EntityManager em, RequestParamsDTO params) {
    CriteriaBuilder countBuilder = em.getCriteriaBuilder();
    CriteriaQuery<Long> countQuery =
countBuilder.createQuery(Long.class);
    Root<Flat> countRoot = countQuery.from(Flat.class);

    Join<Flat, FlatParams> countFlatParams =
```

## Продолжение приложения Г

```
countRoot.join("flatParams");
    List<Predicate> predicates = new ArrayList<>();

    if (params.getMinPrice() != null) {

predicates.add(countBuilder.greaterThanOrEqualTo(countRoot.get("price"),
params.getMinPrice()));
    }
    if (params.getMaxPrice() != null) {

predicates.add(countBuilder.lessThanOrEqualTo(countRoot.get("price"),
params.getMaxPrice()));
    }
    if (params.getNumberOfRooms() != null &&
!params.getNumberOfRooms().isEmpty()) {

predicates.add(countRoot.get("numberOfRooms").in(params.getNumberOfRooms()));
    }
    if (params.getMinSquare() != null) {

predicates.add(countBuilder.greaterThanOrEqualTo(countRoot.get("area").get("total"), params.getMinSquare()));
    }
    if (params.getMaxSquare() != null) {

predicates.add(countBuilder.lessThanOrEqualTo(countRoot.get("area").get("total"), params.getMaxSquare()));
    }
    if (params.getResale() != null && !params.getResale().isEmpty()) {

predicates.add(countFlatParams.get("resale").in(params.getResale()));
    }

    if (params.getWalling() != null && !params.getWalling().isEmpty()) {

predicates.add(countFlatParams.get("walling").in(params.getWalling()));
    }
    if (params.getAddress() != null && !params.getAddress().isEmpty()) {

predicates.add(countBuilder.like(countFlatParams.get("location").get("address"), "%" + params.getAddress() + "%"));
    }

    countQuery.select(countBuilder.count(countRoot))
        .where(countBuilder.and(predicates.toArray(new
Predicate[0])));

    Long result = em.createQuery(countQuery).getSingleResult();
    return result;
}

@Override
public void save(Flat flat) {
    flatRepository.save(flat);
}

public List<Flat> getEvaluatedFlats(Principal principal) {
    User user =
userService.findByEmail(principal.getName()).orElseThrow();
```

## Продолжение приложения Г

```
        return flatRepository.findFlatsByUserId(user.getId());
    }

    @Override
    public List<Flat> getUniqueFlats(ImportanceRequest request) {

        double priceOfOneMark = getPriceOfOneMark(request);
        ImportanceOf importanceOf = getImportanceOfParams(request,
priceOfOneMark);

        List<Object[]> results =
flatRepository.findAverageMarksForEachFlat();
        Map<Long, AverageMarks> averageMarks = getAverageMarks(results);

        Map<Long, Double> flatsWithIntegralPreference =
getIntegralPreferences(importanceOf, averageMarks);

        System.out.println("квартиры и интегральные показатели: " +
flatsWithIntegralPreference);

        HashMap<Long, Double> sorted =
flatsWithIntegralPreference.entrySet().stream()
            .sorted(Map.Entry.<Long,
Double>comparingByValue().reversed())
            .collect(Collectors.toMap(
                Map.Entry::getKey,
                Map.Entry::getValue,
                (e1, e2) -> e1,
                LinkedHashMap::new
            ));

        System.out.println("sorted: " + sorted);

        List<Flat> flats = new ArrayList<>();

        for (Long id : sorted.keySet()) {
            flats.add(getOne(id));
        }

        return flats;
    }

    private Map<Long, Double> getIntegralPreferences(ImportanceOf
importanceOf, Map<Long, AverageMarks> averageMarks){
        Map<Long, Double> flatsWithIntegralPreference = new HashMap<>();

        double integralPreference;

        for (Map.Entry<Long, AverageMarks> flat : averageMarks.entrySet()) {
            AverageMarks avgMark = flat.getValue();
            integralPreference = 0;

            integralPreference += importanceOf.getPrice() *
avgMark.getAvgPrice();
            integralPreference += importanceOf.getNumberOfRooms() *
avgMark.getAvgNumberOfRooms();
            integralPreference += importanceOf.getYear() *
avgMark.getAvgYear();
        }
    }
}
```



## Продолжение приложения Г

```
        integralPreference += importanceOf.getFloor() *
avgMark.getAvgFloor();
        integralPreference += importanceOf.getNumberOfFloors() *
avgMark.getAvgNumberOfFloors();
        integralPreference += importanceOf.getTotalArea() *
avgMark.getAvgTotalArea();
        integralPreference += importanceOf.getLivingArea() *
avgMark.getAvgLivingArea();
        integralPreference += importanceOf.getKitchenArea() *
avgMark.getAvgKitchenArea();
        integralPreference += importanceOf.getWalling() *
avgMark.getAvgWalling();
        integralPreference += importanceOf.getTimeToMetro() *
avgMark.getAvgTimeToMetro();
        integralPreference += importanceOf.getTimeToMall() *
avgMark.getAvgTimeToMall();
        integralPreference += importanceOf.getTimeToClinic() *
avgMark.getAvgTimeToClinic();
        integralPreference += importanceOf.getTimeToKindergarten() *
avgMark.getAvgTimeToKindergarten();
        integralPreference += importanceOf.getTimeToSchool() *
avgMark.getAvgTimeToSchool();
        integralPreference += importanceOf.getDistrict() *
avgMark.getAvgDistrict();
        integralPreference += importanceOf.getAddress() *
avgMark.getAvgAddress();
        flatsWithIntegralPreference.put(flat.getKey(),
integralPreference);
    }
    return flatsWithIntegralPreference;
}

private Map<Long, AverageMarks> getAverageMarks(List<Object[]> results){
    Map<Long, AverageMarks> averageMarks = new HashMap<>();
    for (Object[] row : results) {
        AverageMarks averageMark = new AverageMarks();

        averageMarks.put(((Number) row[0]).longValue(), averageMark);
        averageMark.setAvgPrice( Double.parseDouble(row[1].toString()));
        averageMark.setAvgNumberOfRooms(
Double.parseDouble(row[2].toString()));
        averageMark.setAvgYear( Double.parseDouble(row[3].toString()));
        averageMark.setAvgFloor( Double.parseDouble(row[4].toString()));
        averageMark.setAvgNumberOfFloors(
Double.parseDouble(row[5].toString()));
        averageMark.setAvgTotalArea(
Double.parseDouble(row[6].toString()));
        averageMark.setAvgLivingArea(
Double.parseDouble(row[7].toString()));
        averageMark.setAvgKitchenArea(
Double.parseDouble(row[8].toString()));
        averageMark.setAvgWalling(
Double.parseDouble(row[9].toString()));
        averageMark.setAvgTimeToMetro(
Double.parseDouble(row[10].toString()));
        averageMark.setAvgTimeToMall(
Double.parseDouble(row[11].toString()));
        averageMark.setAvgTimeToClinic(
```

## Продолжение приложения Г

```
Double.parseDouble(row[12].toString()));
    averageMark.setAvgTimeToKindergarten(
Double.parseDouble(row[13].toString()));
    averageMark.setAvgTimeToSchool(
Double.parseDouble(row[14].toString()));
    averageMark.setAvgDistrict(
Double.parseDouble(row[15].toString()));
    averageMark.setAvgAddress(
Double.parseDouble(row[16].toString()));
    averageMarks.put(((Number) row[0]).longValue(), averageMark);
    }
    return averageMarks;
}

private ImportanceOf getImportanceOfParams(ImportanceRequest request,
double priceOfOneMark) {
    return new ImportanceOf(
        priceOfOneMark * request.getPrice(),
        priceOfOneMark * request.getNumberOfRooms(),
        priceOfOneMark * request.getYear(),
        priceOfOneMark * request.getFloor(),
        priceOfOneMark * request.getNumberOfFloors(),
        priceOfOneMark * request.getTotalArea(),
        priceOfOneMark * request.getLivingArea(),
        priceOfOneMark * request.getKitchenArea(),
        priceOfOneMark * request.getWalling(),
        priceOfOneMark * request.getTimeToMetro(),
        priceOfOneMark * request.getTimeToMall(),
        priceOfOneMark * request.getTimeToClinic(),
        priceOfOneMark * request.getTimeToKindergarten(),
        priceOfOneMark * request.getTimeToSchool(),
        priceOfOneMark * request.getDistrict(),
        priceOfOneMark * request.getAddress()
    );
}

private double getPriceOfOneMark(ImportanceRequest request) {
    int sum = 0;
    sum += request.getPrice();
    sum += request.getNumberOfRooms();
    sum += request.getYear();
    sum += request.getFloor();
    sum += request.getNumberOfFloors();
    sum += request.getTotalArea();
    sum += request.getLivingArea();
    sum += request.getKitchenArea();
    sum += request.getWalling();
    sum += request.getTimeToMetro();
    sum += request.getTimeToMall();
    sum += request.getTimeToClinic();
    sum += request.getTimeToKindergarten();
    sum += request.getTimeToSchool();
    sum += request.getDistrict();
    sum += request.getAddress();
    double priceOfOneMark = 1.0 / sum;
    return priceOfOneMark;
}
}
```