

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»
Кафедра IT-инжиниринг

ДОПУЩЕН К ЗАЩИТЕ

Заведующий кафедрой

PhD, доцент

_____ Т.С. Картбаев
« ____ » _____ 2019 г.

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка автоматизированной информационной системы службы КСК

Специальность 5В070300 – «Информационные системы»

Выполнил Омаров Ж.Б. Группа ИС-15-2

Научный руководитель доцент Досжанова А.А.

Консультанты:

по экономической части: к.э.н., доцент _____ *А.И.Бекишева*
« 23 » _____ 05 _____ 2019 г.

по безопасности жизнедеятельности: д.т.н., ст. преп. _____ *Ш.Ш. Бекбасаров*
« 22 » _____ 05 _____ 2019 г.

по применению
программного обеспечения: ст. преп. _____ *М.Н. Майкотов*
« 24 » _____ 05 _____ 2019 г.

Нормоконтролер: ст. преп. _____ *Ж.К. Алимсеитова*
« 30 » _____ 05 _____ 2019 г.

Рецензент: д.т.н., профессор _____ Б. С. Ахметов
« ____ » _____ 2019 г.

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5В070300 – «Информационные системы»

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту Омарову Жандосу Бауыржанұлы

Тема проекта: Разработка автоматизированной информационной системы службы КСК

Утверждена приказом по университету № 124 от «26» октября 2018 г.

Срок сдачи законченного проекта «04» мая 2019 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта):

1. Результаты исследования предметной области;
2. Данные преддипломной практики.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта:

1. Анализ предметной области;
2. Описание исходной ситуации;
3. Обоснование средств и технологий;
4. Проектирование и разработка приложения.

Перечень графического материала (с точным указанием обязательных чертежей): представлены 15 таблиц, 37 иллюстраций.

Основная рекомендуемая литература:

1. Золотов С.Ю. «Проектирование информационных систем», издательство «Эль Контент», 2013 г.
2. Карпова И.П. «Базы данных. Учебное пособие», издательство «Питер», 2013 г.
3. Себеста Р. «Основные концепции языков программирования», издательство «Эксмо», 2011 г.
4. Оппель Эндрю Дж., Вайнберг Пол Н., Грофф Джеймс Р. «SQL. Полное руководство», издательство «Вильямс», 2018 г.

Консультации по проекту с указанием относящихся к ним разделов проекта

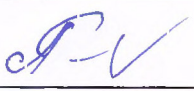
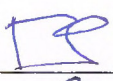
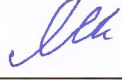
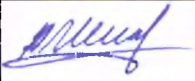
Раздел	Консультант	Сроки	Подпись
Экономическая часть	Бекишева А.И.	02.04.2019- 30.05.2019	
Безопасности жизнедеятельности	Бекбасаров Ш.Ш..	02.04.2019- 30.05.2019	
Программное обеспечение	Майкотов М.Н.	02.04.2019- 30.05.2019	
Нормоконтролер	Алимсеитова Ж.К.	02.04.2019- 30.05.2019	

График
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Основная часть	24.05.2019	Выполнено
Специальная часть	24.05.2019	Выполнено
Ожидаемые технико-экономические показатели	23.05.2019	Выполнено
Безопасность жизнедеятельности	22.05.2019	Выполнено

Дата выдачи задания «26» октября 2018 г.

Заведующий кафедрой _____ Т.С. Картбаев

Научный руководитель проекта  _____ А.А. Досжанова

Задание принял к исполнению студент  _____ Ж.Б. Омаров

Аңдатпа

Дипломдық жобаның тақырыбы: «Кооперативті пәтер иелерінің автоматтандырылған ақпараттық жүйесін құру».

Бұл қорытындылық біліктілік жұмысының мақсаты - осы ақпараттық жүйені өңдеу жолдары қарастырылған, кооперативті пәтер иелерінің автоматтандырылған ақпараттық жүйесіні дайындау. Тәжірибелік бөлімде әр кезеңнің сипаттамасымен ақпараттық жүйелерін әзірлеу скриншоты ұсынылады.

Дипломдық жоба кіріспеден, бес тараудан және қорытындыдан тұрады.

Кіріспеде дипломдық жобаның тақырыбының өзектілігі анықталған, оның мақсаты – қажетті тапсырмаларды жасау және орындау.

Бірінші бөлім тақырыптық аумақты және қолданыстағы аналогтарды талдайды және мәселені анықтайды.

Екінші бөлімде ақпараттық жүйені жасау құралдарын жобалау және таңдау сипатталады.

Үшінші тарауда ақпараттық жүйенің дамуы және міндеттерді практикалық шешудің егжей-тегжейлі сипаттамасы берілген..

Төртінші тарауда әзірленген жобаның экономикалық мүмкіндіктері сипатталады.

Бесінші тарауда жобадағы жұмыс жағдайын жақсарту жолдарына назар аударылады.

Қорытындыда зерттеу және өңдеу нәтижелері көрсетіледі:

Дипломдық жобаның шеңберінде ақпараттық жүйенің дамуы мен материалдарының негізгі кезеңдері зерттелді.

Аннотация

Тема дипломного проекта: «Разработка автоматизированной информационной системы службы КСК».

Целью данного выпускного квалификационного проекта является разработка автоматизированной информационной системы службы КСК, в ходе которой были рассмотрены способы создания данной информационной системы. В практической части представлены скриншоты разработки информационной системы, с описанием каждого этапа.

Дипломный проект состоит из введения, пяти глав и заключения.

Во введении раскрывается актуальность темы дипломного проекта, ставится цель разработки и выполнения необходимых задач.

В первой главе производится анализ предметной области и существующих аналогов, производится постановка задачи.

Во второй главе описывается проектирование и выбор средств разработки информационной системы.

Третья глава описывает разработку информационной системы и подробное описание практического решения поставленной задачи.

В четвертой главе описывается экономическая целесообразность разрабатываемого проекта.

В пятой главе происходит рассмотрение способов по улучшению условий труда в пределах разрабатываемого проекта.

В заключении освещаются результаты исследования и разработки проекта.

В рамках дипломного проекта были изучены материалы и основные этапы разработки информационной системы.

Annotation

Theme of the graduation project: "Development of automated information system service CAO (cooperative apartment owners)".

The purpose of this final qualifying work is to develop an automated information system for the apartment owners cooperative service, during which the ways of creating this information system were considered. In the practical part, screenshots of information system development are presented, with a description of each stage.

The graduation project consists of introduction, five chapters and conclusion.

In the introduction reveals the relevance of the theme of the graduation project, the goal is to develop and perform the necessary tasks.

The first chapter analyzes the subject area and existing analogues, and sets the task.

The second chapter describes the design and selection of information system development tools.

The third chapter describes the development of an information system and a detailed description of the practical solution of the task.

The fourth chapter describes the economic feasibility of the developed project.

In the fifth chapter, consideration is given to ways to improve working conditions within the project being developed.

The conclusion highlights research and development results.

In the context of the graduation project, the materials and the main stages of the development of the information system were studied.

Содержание

Введение	8
1 Исследование предметной области	9
1.1 Анализ предметной области	9
1.2 Анализ существующих аналогов программы	18
1.3 Постановка задачи	22
2 Проектирование и выбор средств разработки	26
2.1 Этапы проектирования	26
2.2 Возможное поле разработки	34
3 Программная реализация	53
3.1 Главная страница	53
3.2 Регистрация	54
3.3 Личный кабинет	55
3.4 Восстановление пароля	56
3.5 Заявка на ремонт	57
3.6 Документация	58
3.7 Информация о кооперативе собственников квартир	58
3.8 Контакты	59
4 Экономическое обоснование разработки проекта	61
4.1 Трудоемкость разработки программного продукта	61
4.2 Расчет затрат на разработку программного продукта	62
4.3 Трудовые ресурсы, задействованные в работе	64
4.4 Определение договорной цены программного продукта	67
4.5 Расчет срока окупаемости программного продукта	67
4.6 Оценка социально-экономических результатов	68
5 Безопасность жизнедеятельности	69
5.1 Анализ условий труда	69
5.2 Расчет тепловых нагрузок в помещении	70
5.3 Расчет внутренних тепловых нагрузок в помещении	72
5.4 Расчет теплового баланса помещения	73
5.5 Выбор кондиционера. Схема расположения	74
5.6 Вывод раздела	75
Заключение	76
Список литературы	77
Приложение А. Техническое задание	79
Приложение Б. Листинг программы	82
Приложение В. Акт внедрения	85

Введение

Кооператив собственников квартир (КСК) – это форма управления имуществом, при которой собственники передают полномочия по управлению домом кооперативу: жилищному, жилищно-строительному, потребительскому, кооперативу собственников помещений (квартир), индивидуальному предпринимателю. Соглашение собственников, принятое на общем собрании, определяет форму управления.

Создать КСК могут собственники многоквартирного дома либо нескольких домов, которые имеют общий двор, систему коммуникаций.

В отличие от управляющей компании, КСК не частная организация, а демократично избранная форма управления домом собственниками квартир. КСК функционирует за счет собранных денег с жильцов на содержание и ремонт дома. При нарушениях жильцы могут обратиться в суд, а председателя КСК можно переизбрать на общем собрании жильцов.

Согласно Закону РК «О жилищных отношениях», КСК считается созданным и приобретает права юридического лица с момента его государственной регистрации или перерегистрации в соответствии с установленным порядком.

В настоящее время в сфере деятельности КСК не прослеживается тенденция к использованию современных интернет-технологий.

Для предоставления населению информации об основных направлениях деятельности конкретного КСК и предлагаемых им услугах будет разработана автоматизированная информационная система (ИС). Это поможет взаимодействию жильцов домов с собственным КСК.

Цель проекта - разработка информационной системы КСК с элементами автоматизированной регистрации и отправок заявок на ремонт.

Для достижения поставленной в проекте цели необходимо решить следующие задачи:

- проанализировать предметную область;
- выполнить обзор и анализ известных аналогов ИС;
- разработать концептуальную модель ИС;
- разработать логическую модель данных ИС;
- разработать физическую модель данных ИС;
- выбрать средства и реализовать ИС и базу данных системы.

1 Исследование предметной области

1.1 Анализ предметной области

1.1.1 Общие сведения о деятельности предприятия

Кооператив собственников квартир «Батыр» является некоммерческой организацией, зарегистрированной в городе Алматы, в Ауэзовском районе. На общем собрании принимается Устав, на основании которого кооператив в будущем будет осуществлять свою деятельность.

Аппаратами управления кооператива являются общее собрание и правление КСК.

КСК «Батыр» – юридическое лицо, имеющее печать со своим названием, расчетный счет в банке и остальные реквизиты.

Официальное название:

– полное название – «некоммерческая организация кооператива собственников квартир Батыр»;

– краткое название – КСК «Батыр».

Местонахождение – г. Алматы, ул. Сулейменова, д. 17.

Основной вид экономической деятельности – управление эксплуатацией жилого фонда.

Главными мотивами создания кооператива собственников квартир являются управление и обеспечения использования общего имущества. Самый высший аппарат управления – общее собрание, исполнительный аппарат – правление, под управлением председателя, контрольный аппарат – ревизор или ревизионная комиссия.

Целевые поступления собственников квартир являются главным источником финансирования деятельности кооператива.

1.1.2 Организационная структура предприятия

В штате работников кооператива состоят три человека: диспетчер, бухгалтер и председатель правления.

Председатель правления является избранным лицом, имеющим организационные полномочиями. На первом заседании правления кооператива, путем голосования членами кооператива на срок, указанный в уставе кооператива, из числа членов правления избирается председатель правления. Согласно регламенту устава КСК, в обязанности председателя может входить бухгалтерский учет. Председатель правления обязан знать служебные обязанности всех работников КСК, должен контролировать их деятельность и требовать выполнения работниками своих служебных обязанностей.

К аппаратам управления кооператива относятся общее собрание членов кооператива и его правление.

Высший орган управления – общее собрание членов. Правление выбирается из членов общего собрания и отчетывается перед ним. Председатель правления является высшим должностным лицом кооператива и

избирается из состава правления кооператива. С целью осуществления ревизии финансовой деятельности КСК избирается ревизор или ревизионная комиссия.

Правление – это исполнительный орган кооператива. В уставе КСК обязана быть четко определена компетенция правления и ее отличие от компетенции общего собрания. За исключением вопросов, отнесенных к исключительной компетенции общего собрания собственников помещений в многоквартирном доме и компетенции общего собрания членов кооператива, правление кооператива имеет право принимать решения по всем вопросам деятельности КСК.

Правление избирается из числа членов кооператива общим собранием членов кооператива на срок, указанный в уставе КСК, и должен длиться не более чем один год. Именно поэтому, через каждый год необходимо организовывать переизбрание правления кооператива. Жилищный кодекс не запрещает лицам, уже избиравшимся в члены правления ранее, становиться членом правления в очередной раз. Во главе правления находится председатель правления кооператива. Устав кооператива должен определять конкретные сроки и периодичность созыва заседаний правления. В архиве кооператива, отдельно от протоколов общих собраний, должны храниться протоколы, которые ведутся в ходе заседания. При наличии большинства членов правления набирается необходимый кворум, установленный для заседания правления.

В кооперативе бухгалтер относится к административно-управленческому органу. В своей работе он руководствуется законодательством РК, подзаконными актами по организации бухгалтерского и финансового учета для кооператива, его уставом, решениями общих собраний, распоряжениями правления и указами председателя правления, при условии, что они не противоречат законодательству.

Члены ревизионной комиссии избираются по необходимости. На общем собрании членов кооператива происходит процесс избрания членов ревизионной комиссии. Если количество членов кооператива не превышает 50 человек – достаточно избрать одного ревизора, а в случае если количество членов кооператива большое 40-60 человек, следует избрать ревизионную комиссию. Члены ревизионной комиссии в праве избрать на заседании своего председателя, который будет возглавлять ее и в дальнейшем организует ее работу.

В жилищном кодексе говорится о запрете избрании одних и тех же лиц в правление и ревизионную комиссию КСК.

Организационная структура КСК «Батыр» представлена на рисунке 1.1.

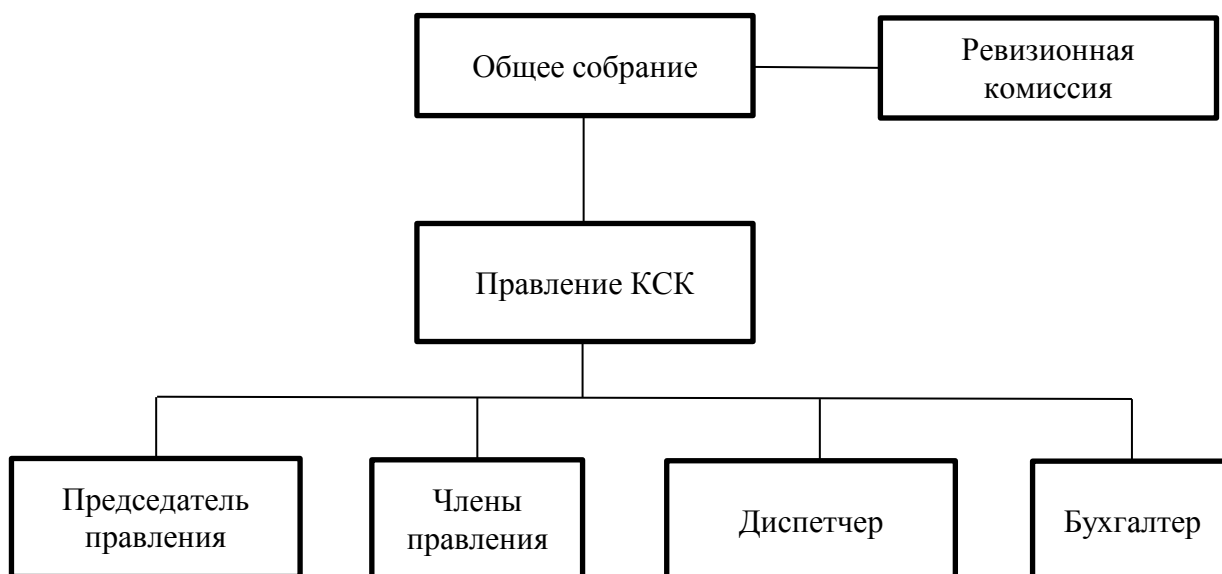


Рисунок 1.1 – Организационная структура КСК «Батыр»

Ревизор или ревизионная комиссия наделены следующими полномочиями:

- не реже одного раза в год проводить проверку финансовой деятельности кооператива;
- демонстрировать общему собранию кооператива заключение о смете расходов и доходов, отчет размерах обязательных взносов и платежей и финансовой деятельности;
- докладывать о своей деятельности перед общим собранием.

1.1.3 Обязанности кооператива собственников квартир

В обязанности кооператива входят работа по текущему ремонту и обслуживанию сетей отопления, водопроводов, канализации, электротехнических сетей. Так в текущий ремонт включается починка текущей крыши, ремонт в подъезде, замена стекол, дверей, перил, герметизация стыков, заделка выбоин и трещин в стенах. Обязаны раз в пять лет оштукатурить и покрасить стены, потолки, лестницы в подъезде. Также по заявкам жильцов заделываются трещины в стенах и заделываются выбоины. При этом полная замена крыши в текущий ремонт не включается, только ремонт и устранение течи. В текущий ремонт также входят все виды работ по устранению течи кровель кроме полной замены, все виды штукатурно-малярных работ во всех помещениях минимум один раз в пять лет.

При неисправности отопления проводят ремонт, осенью и весной включают и отключают тепло, весной проверяют состояние системы, регулируют ее при запуске. Также КСК устраняют небольшие неисправности: очистка арматуры и грязевиков от накипи и солевых отложений, устранение

течи, закрепление участков и опор под расшатавшимися трубопроводами. По необходимости КСК делает теплоизоляцию трубопроводов и их гидравлическую промывку. Осматривают индивидуальные тепловые пункты два раза в месяц с замером температуры давления, коэффициента смещения

В отношении систем водоснабжения служба КСК обязана два раза в месяц проверить тепловые пункты и замерить температуру давления. В начале и конце отопительного сезона служба переключает систему на режим ГВС (горячего водоснабжения), заменяет прокладок запорной и регулирующей арматуры, очищает их от накипи и солей, устраняет течь. Также если какие-то отдельные участки до одного метра длиной повреждены, то служба КСК может их заменить. Также служба заменяет дроссельные органы и укрепляет расшатавшиеся опоры.

В отношении канализации КСК должен: прочистить ее в подвале, устранить течь, везде, где она есть на резьбе, набить на вентили сальники и заварить все отверстия. Также при необходимости из подвала откачивается вода, заменяются участки водоснабжения в перекрытии. КСК также должны вызвать службы дезинфекции, дезинсекции, дератизации подвала.

В компетенции КСК находится и обслуживания электропроводок общего назначения: заменяются сгоревшие предохранители в главном щите; следят за наличием освещения в подъездах – при этом лампочки заменяются только на первом этаже и крыльце (если оно там есть); вызывают аварийно-диспетчерские службы (по договору или заявкам); восстанавливаются обгоревшие провода фазы-нуль в главном щите и в этажных щитах; подается напряжение и заменяются выключатели дежурного света.

К административным обязанностям КСК относятся организация проведения собраний, письменных опросов собственников квартир, в том числе рассмотрение обращений и жалоб по качеству обслуживания, ведение документаций налогового, финансового, бухгалтерского и статистического учета и отчетности, а также технических документаций, связанных с содержанием и управлением объекта кондоминиума в соответствии с жилищным законодательством Республики Казахстан. Кооперативы должны уведомлять о проведении технических работ, в том случае если они препятствуют жильцам использовать хотя бы одну из коммуникаций дома. В соответствии с заключенными договорами обязаны открывать текущие и сберегательные счета на непосредственно объект кондоминиума, финансировать услуги и работы подрядчиков, субъектов сервисных деятельности (участников процесса производства работ и услуг). КСК занимаются регистрационным учетом арендаторов и собственников квартир.

Кроме вышеописанных обязанностей, кооператив также убирается в придомовой территории. В отношении уборки, кооператив занимается уборкой льда с дорожек, вывозом снега, засыпанием песка на скользкую дорогу, монтированием отводов для дождя от дома, уборкой мусора и листьев с крыши, его вывозом.

Помимо основных обязанностей кооператив имеет право оказывать дополнительные услуги, связанные с благоустройством района, двора или дома. На оказание этих услуг кооператив вправе проводить сбор денежных средств, размер которых обсуждается на общем собрании жильцов дома. Законодательство Республики Казахстан рассматривает такие сборы как накопление денежных средств в банках второго уровня на проведение капитального или текущего ремонта, согласно статье 31, пункту 8 Закона РК «О жилищных отношениях».

КСК обязаны на каждый объект кондоминиума открывать в банках текущие счета, на который жильцы ежемесячно перечисляются платежи в размере, который определяется на собрании собственников квартир, но его размер не должен быть менее 0,02-кратного месячного расчетного показателя (МРП).

Для прозрачности работы кооператива, председатель правления обязан не реже одного раза в квартал представлять отчет об использовании денежных средств по запросу собственников квартир. На общем собрании жильцов дома объявляются итоги отчета.

Жильцы вправе потребовать Акт проверки финансово-хозяйственной деятельности кооператива от ревизионной комиссии. Ревизионная комиссия дает заключение по ежегодному отчету правления КСК и по представленной правлением годовой смете расходов и доходов.

Для выполнения всех вышеописанных обязательств кооператив составляет договор о сотрудничестве с аварийно-диспетчерскими службами(АДС).

Аварийно-диспетчерская служба следит за выполнения заявок жителей по устранению мелких неисправностей, повреждений домового оборудования и работой инженерного оборудования жилых домов. В задачу АДС также входит принятие оперативных мер по устранению аварийных ситуаций с системами канализации, теплоснабжения, воды и электричества. АДС направляет аварийные бригады на ликвидацию аварийных ситуаций в любое время дня и день недели.

Для повышения эффективности взаимодействия жильцов со службой КСК необходимо усовершенствовать ее деятельность на основе современных интернет-технологий.

1.1.4 Обзор информационных систем и технологий

На данный момент в КСК «Батыр» используются два персональных компьютера для председателя и бухгалтера. Оба компьютера имеют доступ в интернет для взаимодействия с интернет банком. В таблице 1.1 приведена характеристика каждой из рабочих станций.

Таблица 1.1 – Конфигурация компьютера председателя

Процессор	Intel Celeron 2900 MHz
-----------	------------------------

Опер. Память	4 Gb
Винчестер	Western Digital Caviar Blue WD10EZEX (1000 Гб)

Продолжение таблицы 1.1

Флоппи	Отсутствует
Привод	LITE-ON IHAS124
Видеокарта	Intel HD Graphics 610
Монитор	Samsung Sync Master B2030
Сетевая карта	Имеется

Таблица 1.2 – Конфигурация компьютера бухгалтера

Процессор	Intel Celeron 2900 MHz
Опер. Память	4 Gb
Винчестер	Western Digital Caviar Blue WD10EZEX (1000 Гб)
Флоппи	Отсутствует
Привод	LITE-ON IHAS124
Видеокарта	Intel HD Graphics 610
Монитор	Acer AL1917
Сетевая карта	Имеется

В КСК «Батыр» имеются все необходимые лицензионные программные продукты: операционные системы (ОС) Microsoft Windows 7 Professional, офисные программы Microsoft Office 2010, антивирусная программа Kaspersky Internet Security 2015 с ежедневно обновляемыми базами вирусов, 1С: Предприятие 8.3.

Бухгалтер использует следующее специализированное программного обеспечение:

1. 1С Предприятие 8.3 – Налоговый и бухгалтерский учет; расчет амортизации и учет основных средств; формирование бухгалтерской, налоговой и другой отчетности в различные органы; бухгалтерский учет и контроль смет расходов бюджетных организаций в соответствии с ведомственными инструкциями и законодательством;

2. налогоплательщик ЮЛ - предназначена «для автоматизации процесса ввода налоговой и бухгалтерской отчетности, печати документов на бумажных носителях, формирования введенной информации бухгалтерской и налоговой отчетности на магнитных носителях налогоплательщиком и передачи отчетности в налоговые органы в электронном виде записанные на магнитных носителях».

На основе имеющихся данных установлено, что информационная система в КСК «Батыр» не нуждается в локальной сети, так как имеющееся специализированное программное обеспечение работает в локальном режиме, и для нормального функционирования локальная сеть не нужна. Так как малый объем данных легко контролируется локально установленным антивирусным программным обеспечением, имеющаяся система безопасности полностью удовлетворяет потребностям информационной защиты КСК «Батыр».

После анализа конфигурации существующих рабочих станций, было установлено, что они удовлетворяют необходимым требованиям для функционирования будущей информационной системы, так как они уже имеют широкополосный выход в сеть Интернет.

1.1.5 Концептуальное проектирование

Концептуальная модель – это отраслевая модель, которая разрабатывается для предприятия внешним консультантом на базе методологии референтных моделей и реинжиниринга бизнес-процессов, предложенных непосредственно поставщиками систем управления предприятиями общественного питания[3].

В соответствии со стандартами ISO 9001:2000 и ERP концептуальная модель подразумевает унификацию и совершенствование основных процессов предприятия.

На этапе концептуального проектирования информационных систем применяется методология реинжиниринга бизнес-процессов предприятий, с целью их оптимизации и совершенствования.

Стадии цикла совершенствования:

- описание способов реализации процессов предприятия, используемых на данный момент (бизнес-процесс «как есть»). Выявление преград для развития предприятия;

- описание желательных вариантов исполнения процессов компании с использованием передового опыта (стандарты ISO, ERP). Нахождение путей для преодоления препятствий и изменений в процессах, где препятствия не проявляются, но настоящая деятельность не отвечает передовому опыту (бизнес-процесс «как должно быть»);

- обеспечение исполнения на предприятии желаемых вариантов реализации бизнес-процессов с поддержкой современных информационных технологий. Ввод исполнительных механизмов, которые обеспечивают рост эффективности изучаемого бизнес-процесса.

Рекомендуется применить следующие методологии структурного проектирования и анализа для разработки концептуальной модели бизнес-процесса:

- IDEF0 (ICAM Definition) – это графическая нотация, рассчитанная для формализации и описания бизнес-процессов и методология функционального моделирования;

- DFD (от англ. data flow diagrams – «диаграммы потоков данных») – это методология структурного анализа, которая описывает хранилища данных, к которым осуществляется доступ, потоки данных, логические функции, адресаты данных и внешние по отношению к системе источники.

Для начала необходимо построить контекстную диаграмму бизнес-процесса в IDEF0, после чего ее декомпозируют средствами методологии DFD.

1.1.6 Разработка и анализ модели бизнес-процесса «как есть»

На сегодняшний день бизнес-процесс управления маркетингом в КСК «Батыр» не автоматизирован. Он состоит из рассылки писем об услугах и мероприятиях организации и расклейки объявлений.

Бизнес-процесс контроля заявок является обеспечивающим для бизнес-процесса оказания жилищно-коммунальных услуг кооператива.

Бизнес-процесс контроля заявок:

- клиент (жильец) звонит в КСК с заявлением об оказании услуги, подразумевающее техническое обслуживание жилых домов и инженерной инфраструктуры;

- диспетчер принимает заявления, уточняет у клиента категорию и срочность услуг;

- диспетчер составляет заявку на обслуживание на основании заявления жильца;

- диспетчер сохраняет данные заявки в журнал учета заявок, который разработан на базе книги табличного процессора (ТП) Microsoft Excel;

- диспетчер информирует жильца по контактному телефону о дате и времени прихода ремонтной бригады.

Любые действия КСК «Батыр» регламентируются уставом.

Модель «как есть» иллюстрирует используемые в настоящий момент принципы организации бизнес-процесса управления заявками и маркетингом КСК «Батыр».

На рисунке 1.2 показана контекстная диаграмма методологии «как есть» управления маркетингом КСК «Батыр», разработанная в методологии IDEF0, с точки зрения управляющего.

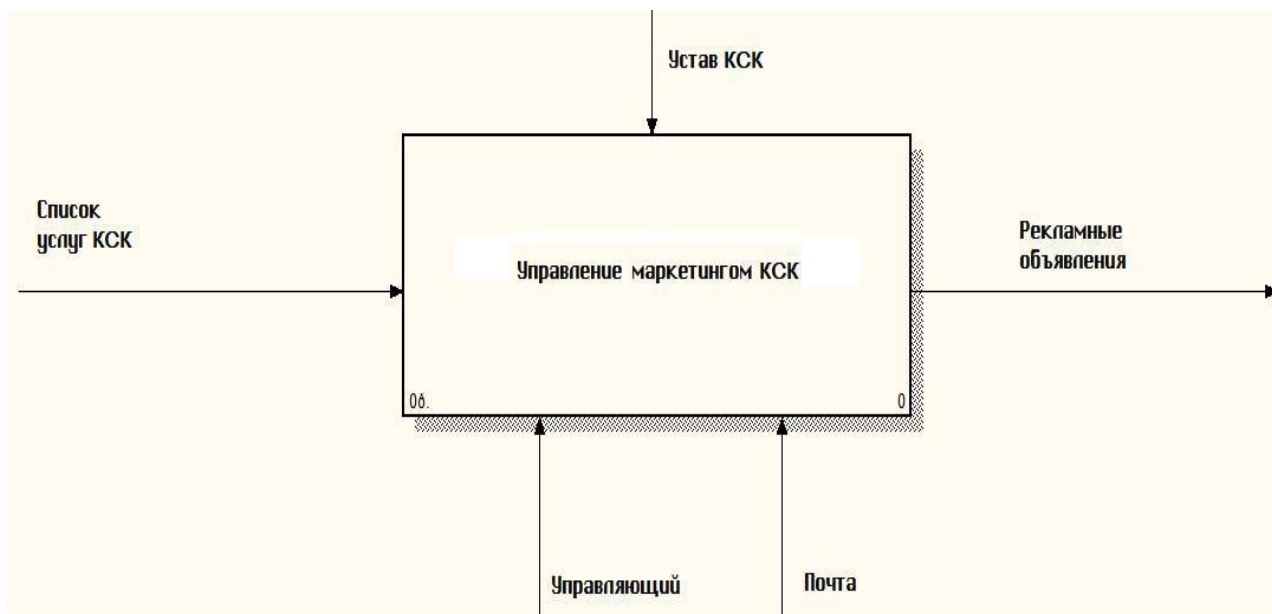


Рисунок 1.2 – Контекстная диаграмма маркетинга в кооперативе «как есть» в методологии IDEF0 (0-й уровень)

На рисунках 1.3 и 1.4 представлена модель «как есть» управления заявками диспетчером, созданная в методологиях IDEF0 и DFD.

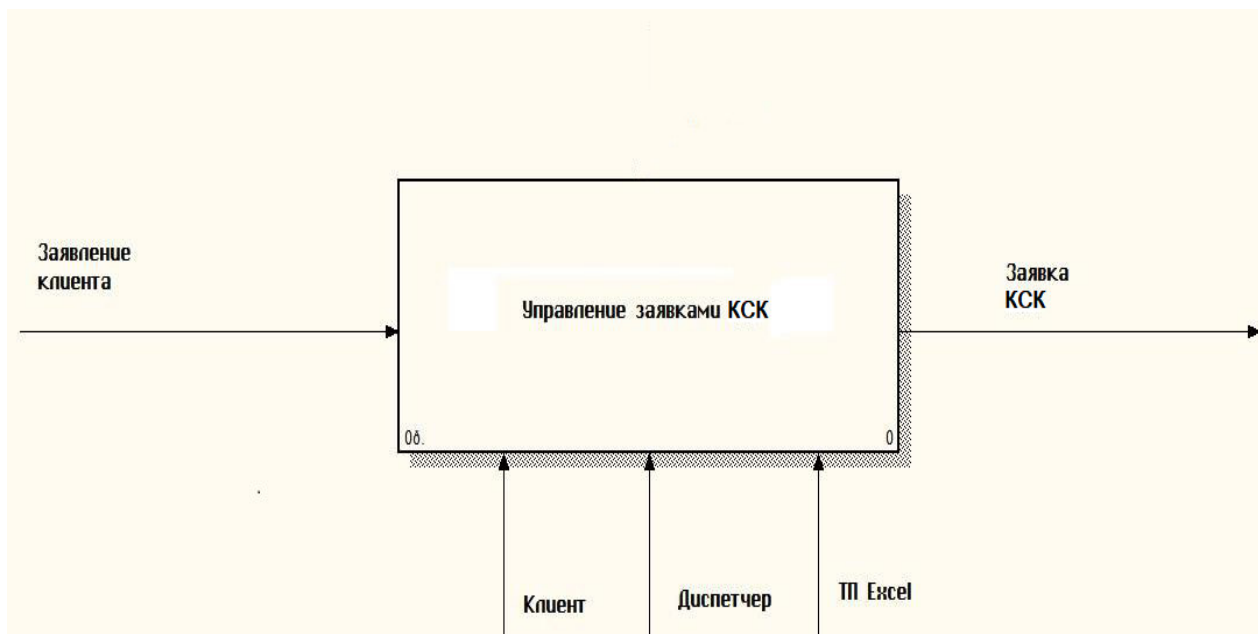


Рисунок 1.3 – Контекстная диаграмма управлением заявками кооператива «как есть» в методологии IDEF0 (0-й уровень)

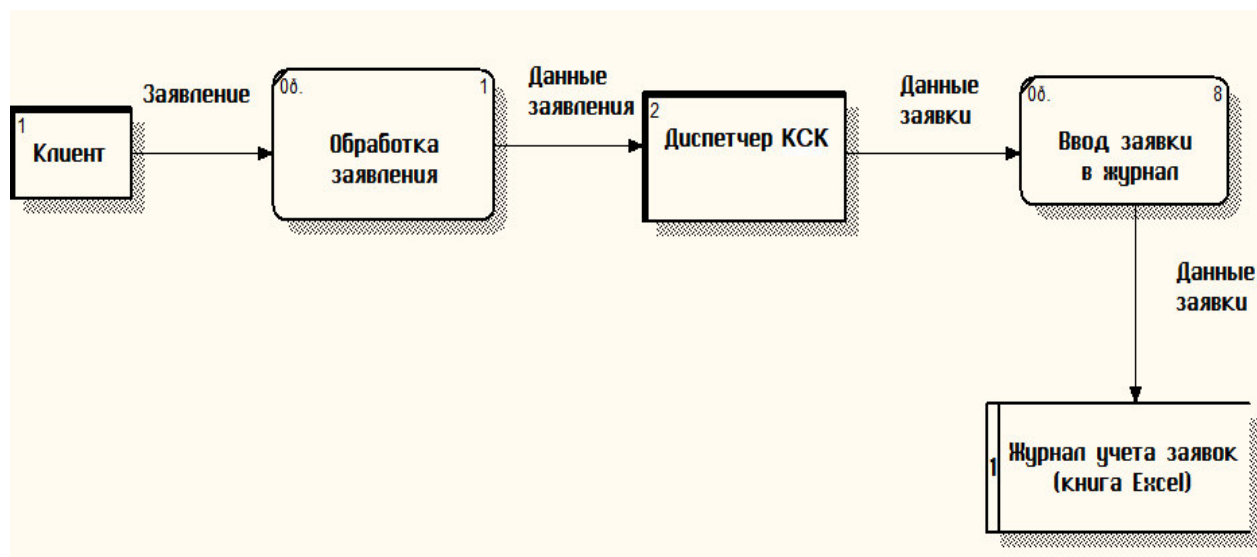


Рисунок 1.4 – DFD-декомпозиция управлением заявками кооператива «как есть» (1-й уровень)

Показанные диаграммы состоят из следующие элементы:

- входные данные: заявление клиента, список услуг;
- выходные данные: заявка, рекламные объявления;
- управляющие воздействия: устав кооператива;
- исполнители: клиент, диспетчер, управляющий, ТП Excel, почта.

Бизнес-процессы управления заявками и маркетингом кооператива будут проанализированы и усовершенствованы на основе данной модели «как есть».

Анализ показал, что бизнес-процессы кооператива имеют следующие недостатки:

- для жильцов отсутствует возможность ознакомиться с новостями, услугами и документами кооператива с помощью сети Интернет;
- жильцы не имеют возможности оформления заявки на ремонт в режиме онлайн и не могут отследить состояние заявки;
- ТП Excel не имеет достаточной функциональностью для управления заявками для ведения журнала на его базе.

С учетом всего принято решение усовершенствовать бизнес-процесс управления заявками и маркетингом путем разработки и внедрения ИС КСК.

1.2 Анализ существующих аналогов программы

В настоящее время в интернете отсутствуют какие-либо представительства КСК, поэтому в целях анализа выбраны российские аналоги КСК – товарищества собственников жилья(ТСЖ).

ТСЖ «Родники-2» (<http://tsj-rodniki-2.ru>). На рисунке 1.5 представлена главная страница сайта. На нем имеется централизованное информирование жильцов, а также всех заинтересованных лиц следующей информации:

- общая информация об товариществе собственников жилья;

- основные показатели финансово-хозяйственной деятельности товарищества собственников жилья;
 - сведения о выполняемых работах и оказываемых услугах по ремонту общего имущества и его содержанию в многоквартирном доме;
 - условия и порядок предоставления услуг по ремонту и содержанию общего имущества в многоквартирном доме;
 - информация о ценах на работу и услуги по ремонту и содержанию общего имущества в многоквартирном доме;
 - информация о стоимости и тарифах на коммунальные ресурсы.
- Жилец имеет возможность подавать заявки на ремонт в личном кабинете после прохождения регистрации.

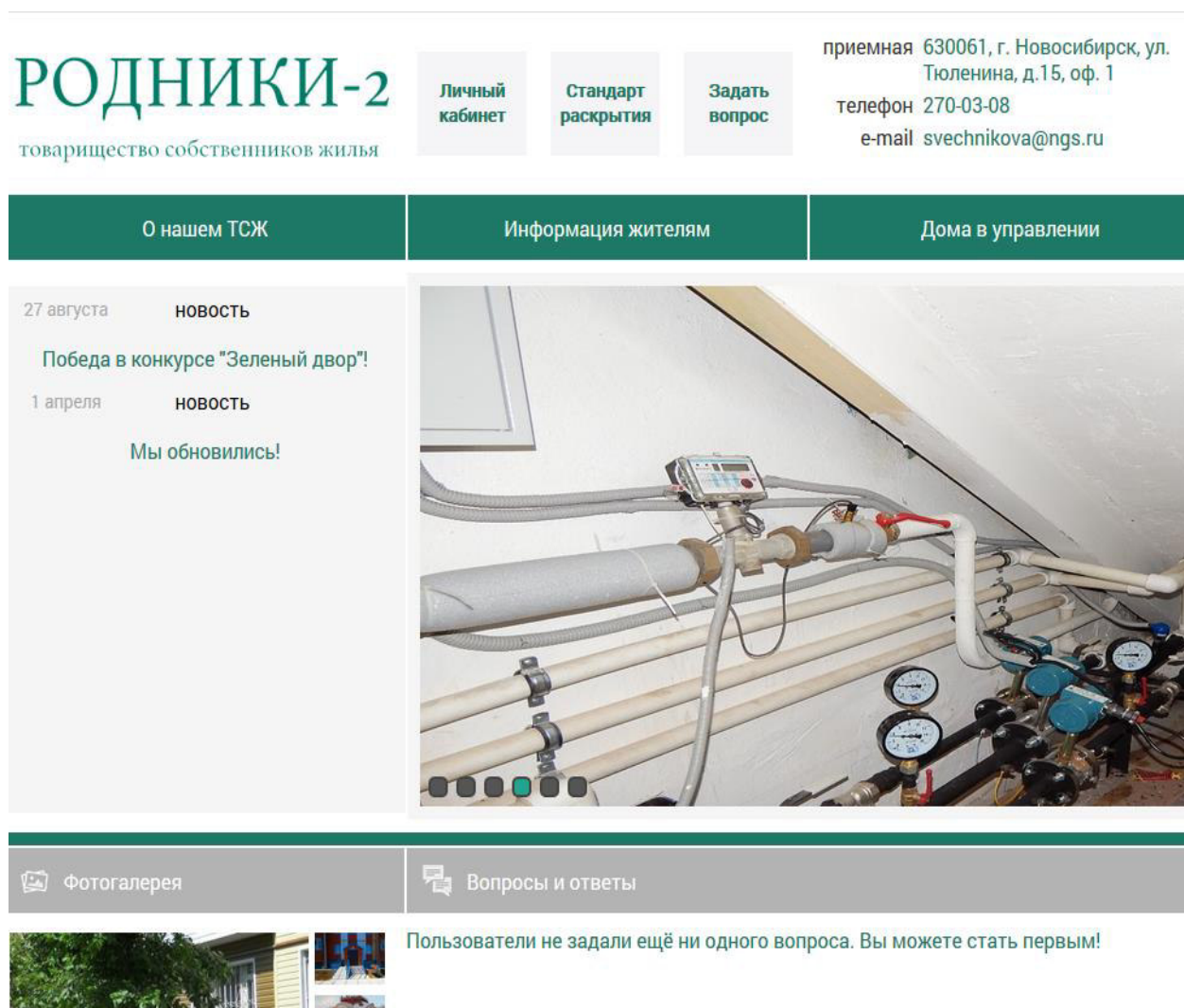


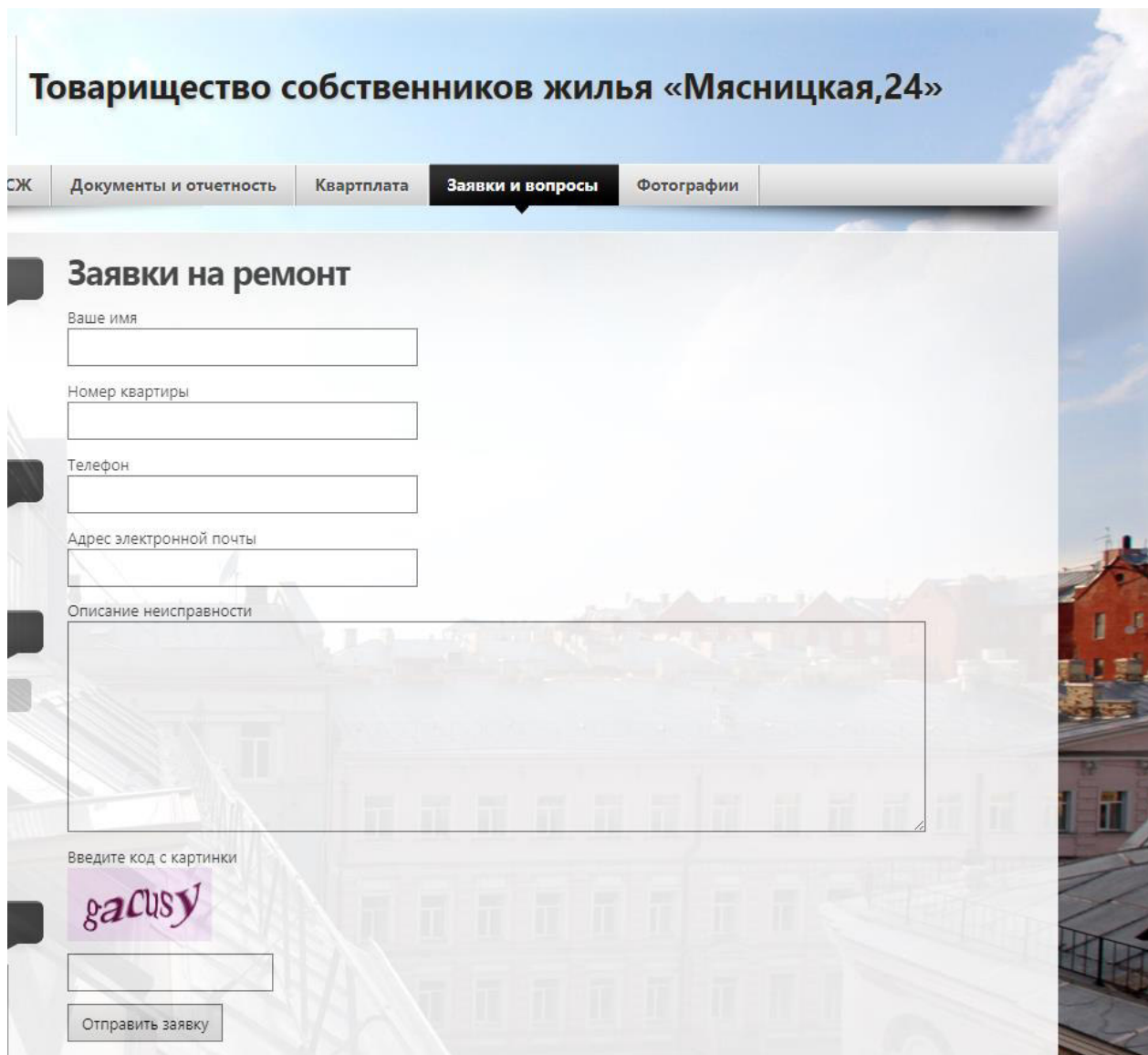
Рисунок 1.5 – Скриншот главной страницы ТСЖ «Родники-2»

ТСЖ «Мясницкая,24» (<http://myasnitskaya24.ru>). Сайт предназначен для предоставления сведений о деятельности ТСЖ, квартплаты, тарифов и услуг. Хранит в открытом доступе документы и отчетность за все время работы сайта. Сайт приобретен как готовый продукт, который требует ежегодной

оплаты за стоимость обслуживания, включающая регистрацию доменного имени сайта, техническое обслуживание, обновление версий сайта и его хостинг.

Предоставляется информационная поддержка следующих услуг:

- оплата коммунальных услуг;
- вопросы к бухгалтеру или к правлению ТСЖ;
- заявки на ремонт (рисунок 1.6).



The screenshot shows the website interface for the apartment owners' association «Мясницкая, 24». The main navigation bar includes links for «СЖ», «Документы и отчетность», «Квартплата», «Заявки и вопросы» (highlighted), and «Фотографии». The page title is «Товарищество собственников жилья «Мясницкая, 24»». The main content area is titled «Заявки на ремонт» and contains a form with the following fields: «Ваше имя», «Номер квартиры», «Телефон», «Адрес электронной почты», and «Описание неисправности». Below the description field is a CAPTCHA image with the word «gacusy» and a text input field for the code. At the bottom of the form is a button labeled «Отправить заявку».

Рисунок 1.6 – Скриншот страницы отправки запроса на ремонт в ТСЖ «Мясницкая, 24»

Альтернативные способы подачи заявок:

- в устной форме непосредственно работнику товарищества;
- в разговоре по телефону с управляющим по дому;
- по электронной почте.

Жильцы могут использовать свой личный кабинет для интерактивной связи с ТСЖ.

ТСЖ «Ленина» (www.tsg-lenina.ru). Сайт предоставляет сведения о деятельности товарищества и организации обратной связи с АДС. На рисунке 1.7 показано окно диспетчера заявок.



Диспетчер заявок

* Поля со * заполнять обязательно!!!

Имя, Фамилия*

Ваш Email адрес*

Напишите, кому хотите сделать заявку*

Опишите Вашу проблему*



Код CAPTCHA*

Рисунок 1.7 – Страницы отправки заявок в ТСЖ «Ленина, 28»

На сайте отсутствует возможность интерактивной связи с жильцом через личный кабинет.

В таблице 1.3 проведен сравнительный анализ характеристик исследованных аналогов.

Таблица 1.3 - Сравнительный анализ представительств ТСЖ

Параметр/Аналог	ТСЖ	ТСЖ	ТСЖ
-----------------	-----	-----	-----

	«Никольское»	«Мясницкая,24»	«Ленина»
Режим работы 7/24/365	+	+	+
<i>Продолжение таблицы 1.3</i>			
Легкость в управлении контентом	+	+	+
Личный кабинет	+	+	-
Поддержка популярных браузеров	+	+	+
Управление заявками в режиме онлайн	+	+	+
Низкая затраты на разработку и ввод в эксплуатацию	-	-	+
Баллы	5	5	5

Из таблицы 1.3 следует, что представленные аналоги не отвечают всем требованиям, предъявляемым к ИС КСК.

С учетом вышеизложенного принято решение о разработке новой ИС КСК, который будет удовлетворять нашим требованиям.

1.3 Постановка задачи

Рассмотренные известные ИТ-решения, их анализ показал, что ни одно из рассмотренных программных средств полностью не удовлетворяет требованиям разрабатываемой автоматизированной информационной системы КСК.

Поэтому было принято решение о разработке новой автоматизированной ИС КСК.

Для разработки, автоматизированной ИС КСК выбрана методология бизнес-моделирования, опирающаяся на трехуровневое представление проектируемой информационной системы: концептуальный, логический и физический[1].

Отсутствует автоматизированная проверка при регистрации жильца.

Выявлен основной недостаток бизнес-процесса управления заявками – отсутствие у клиента возможности вводить заявки онлайн.

Усовершенствование указанного бизнес-процесса достигается путем разработки и внедрения ИС КСК.

1.3.1 Разработка модели бизнес-процесса «как должно быть»

Структурно-функциональная модель «как должно быть» представляет собой концептуальную модель усовершенствованного бизнес-процесса и является основой технического задания на создание или модернизацию информационной системы.

Принимая во внимание вышеперечисленные недостатки существующих бизнес-процессов, на рисунках 1.8 и 1.9 представлена разработанная модель бизнес-процессов управления заявками и маркетингом КСК именуемая «как должно быть».

На представленных ниже диаграммах бизнес-процессов имеются следующие элементы:

- элементы входных данных: данные аккаунта клиента, онлайн-заявка, список объявлений;
- элементы выходных данных: электронная заявка, контент сайта;
- элементы управляющего воздействия: устав КСК;
- элементы исполнителей: администратор, сайт КСК, клиент.

Красным цветом выделены новые и измененные элементы.

Из этого следует, что усовершенствование исследуемого бизнес-процесса можно достичь путем разработки и внедрения отвечающего требованиям ИС КСК.

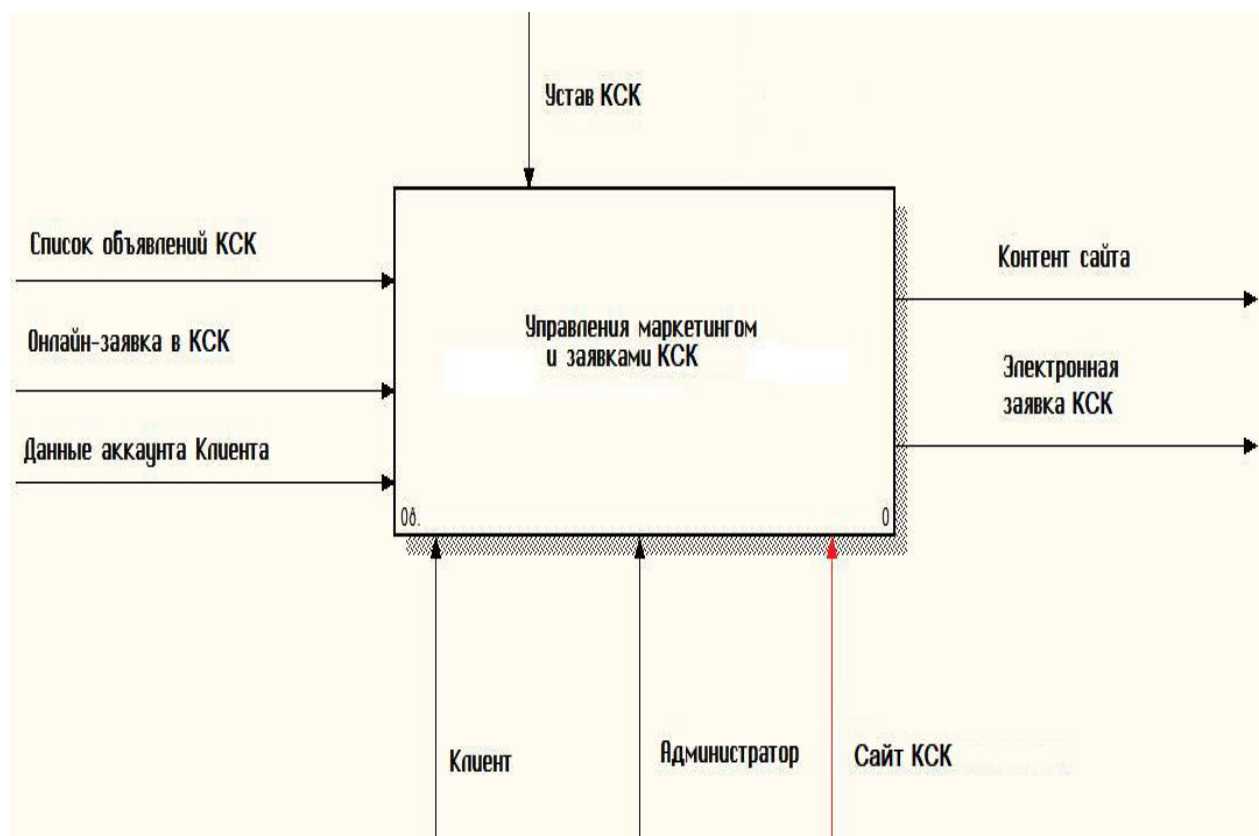


Рисунок 1.8 – Контекстная диаграмма управления заявками и маркетингом КСК «как должно быть» (0-й уровень)

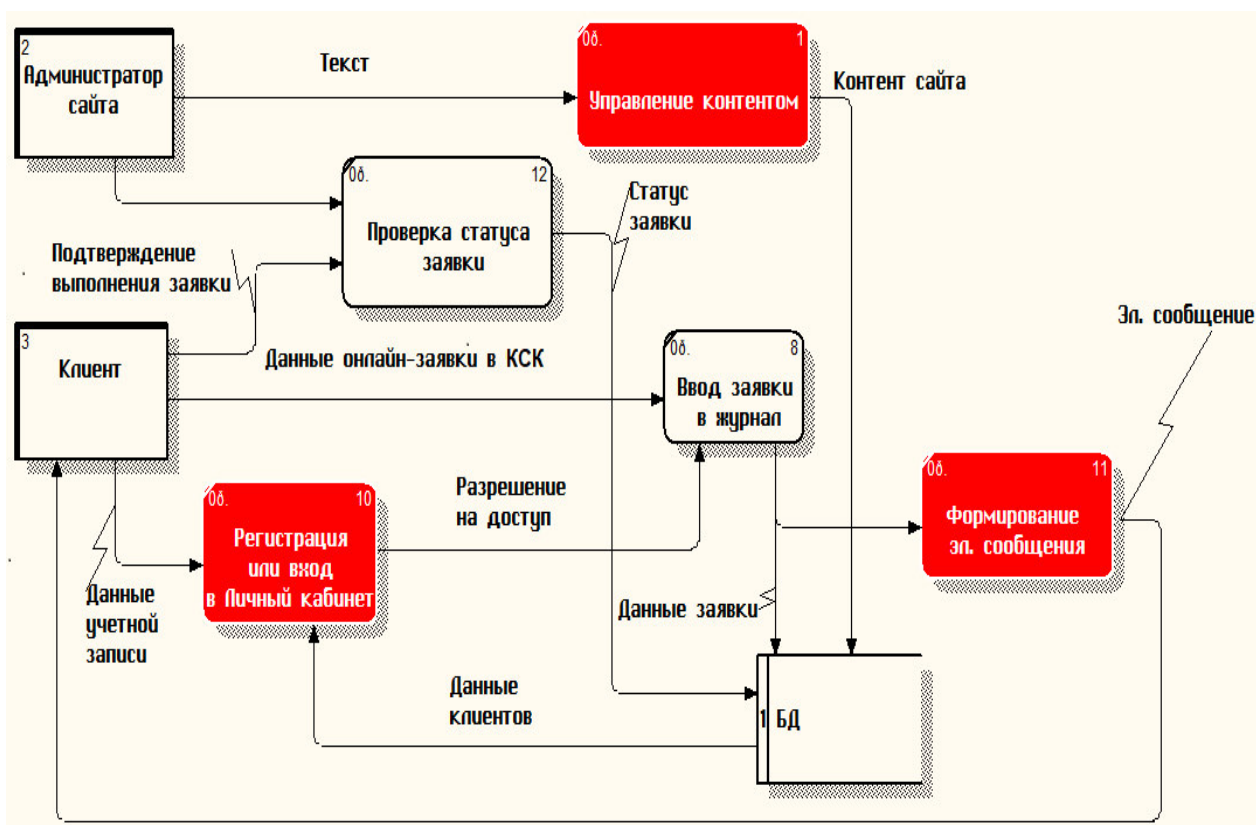


Рисунок 1.9 – DFD-диаграмма управления заявками и маркетингом КСК «как должно быть» (1-й уровень)

1.3.2 Требования к информационной системе

В результате анализа предметной области и поставленных задач, эти функции и требования будут реализованы в виде веб-сайта.

Сайт должен будет представлять собой информационную структуру, доступную в сети интернет в любое время дня и день недели, под доменным именем kskbatyr.kz.

Сайт должен состоять из следующих взаимосвязанных разделов:

- главная страница;
- информация о КСК;
- документация;
- заявка на ремонт;
- контакты.

Для поддержки сайта и управления сайтом от персонала не должно требоваться специальных технических навыков, знания технологий или программных продуктов, только общие навыки работы с компьютером и веб-браузером.

Информация на сайте будет доступна для всех. Жильцы должны иметь личный кабинет, в котором будут отчеты по домам, скачать которые могут только они.

Регистрация на сайте должна быть доступна только для жильцов домов, обслуживаемых КСК «Батыр». Для этого пользователем будет вводиться номер договора купли-продажи квартиры, который должен иметься в базе данных.

Стиль сайта следует сделать деловым и современным. Рекомендуется использовать белый и голубой цвета в качестве фона сайта.

Решение о выборе средств разработки остается за разработчиком.

Информация и навигация на сайте должна быть выполнена на русском языке.

Требования для программного обеспечения клиентской части:

- поддержка популярных браузеров;
- быстрый доступ ко всем страницам сайта.

Сайт должен функционировать на любых персональных компьютерах, а стоимость работ по разработке и вводу сайта в эксплуатацию должна быть низкой.

2 Проектирование и выбор средств разработки

2.1 Этапы проектирования

Имеется большое количество технологий и инструментальных средств, с помощью которых можно создать в оптимальный проект информационной системы, начиная со стадии анализа и заканчивая стадией создания программного кода системы. В основном эти технологии имеют весьма жесткие требования к используемым ресурсам и процессу разработки, а попытки изменить их в конкретные проекты, как правило, оказываются безуспешными. Данные технологии представлены CASE-средствами, одним из которых является язык объектно-ориентированного моделирования Unified Modeling Language (UML).

Логическая и объектная модели данных веб-сайта являются результатом логического моделирования.

Достаточно построить диаграммы UML для описания объектной модели информационной системы на логическом уровне, отображающие динамический, статический и функциональный аспекты проектируемой информационной системы.

На рисунке 2.1 изображена диаграмма деятельности ИС КСК, отображающая алгоритм его работы.

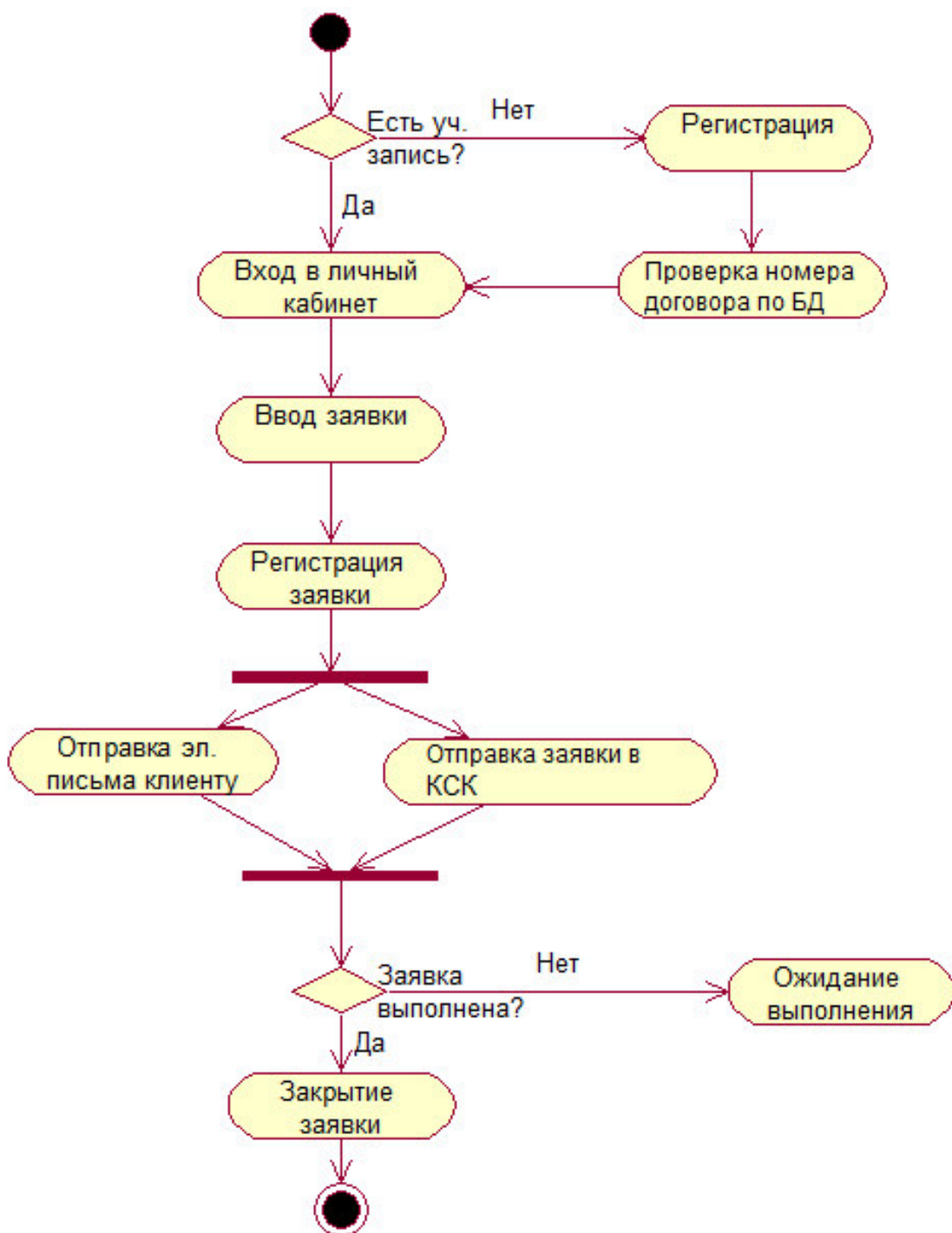


Рисунок 2.1 – Диаграмма деятельности ИС КСК

2. Диаграмма вариантов использования бизнес-процесса управления заявками

Построим диаграмму вариантов использования бизнес-процесса «как должно быть» для отображения функционального аспекта системы.

Диаграммы вариантов использования (use case diagram) применяются как при моделировании функциональных требований к проектируемой

системе при ее проектировании и разработке, так и при бизнес-анализе для моделирования видов работ, выполняемых организацией.

Диаграмма вариантов использования характеризует функциональные возможности анализируемой системы моделью «как должно быть», предоставляя дополнительную информацию об отношениях между внешними пользователями-актерами и разными вариантами использования. Помимо этого, диаграмма позволяет найти границы рассматриваемой информационной системы.

Варианты применения проявляются только в терминах того, как они выражаются, когда анализируются внешним пользователем, при этом не описывают, какие функциональные возможности даны внутри информационной системы.

Диаграмма вариантов использования бизнес-процесса управления заявками «как должно быть» показана на рисунке 2.2.

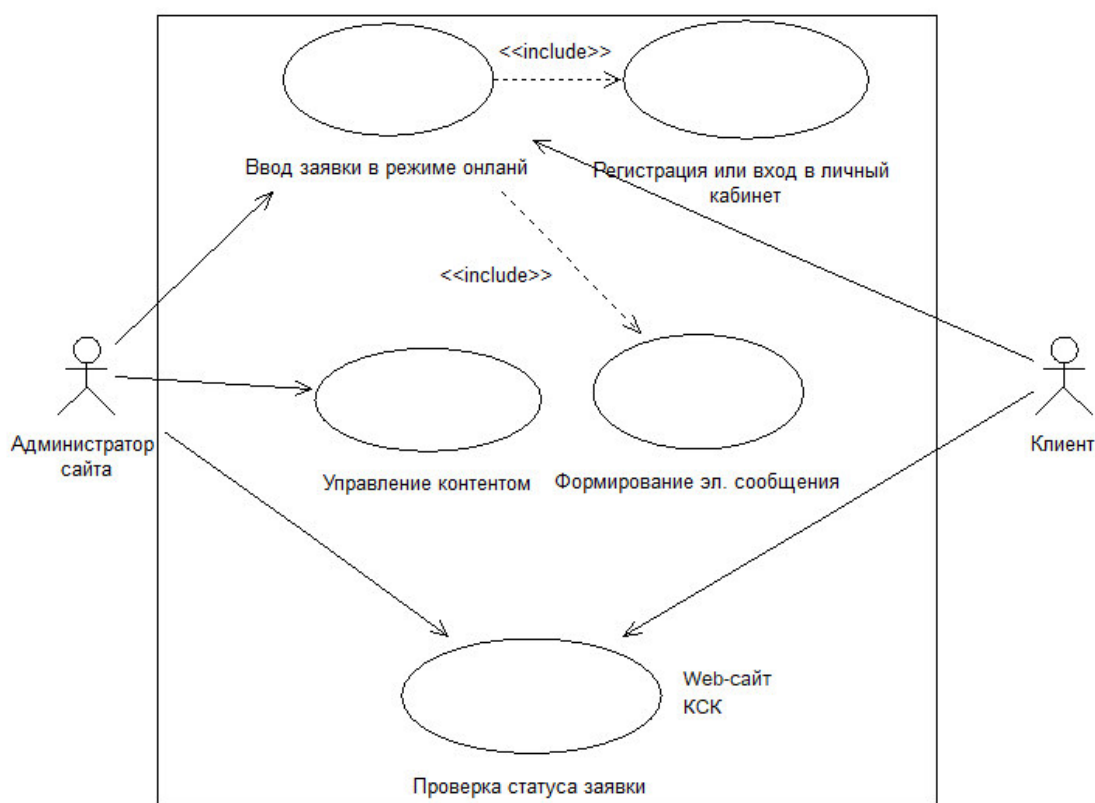


Рисунок 2.2 – Диаграмма вариантов использования процесса управления заявками «как должно быть»

Актеры:

- клиент – пользователь ресурса (житель), имеющий возможность ввода заявки;
- администратор сайта – зарегистрированный пользователь ресурса, который обладает правами администратора сайта и управляющий всеми данными и контентом сайта.

Варианты использования:

- непосредственно ввод заявки в журнал – клиент может ввести заявку на ремонт;
- регистрация или вход в личный кабинет (включение) – незарегистрированный пользователь может пройти регистрацию. При прохождении регистрации пользователь (клиент) имеет почту и пароль, которые он будет использовать для входа на сайт;
- создание электронного сообщения (включение) – жильцу по электронной почте отправляется подтверждение об успешной отправке заявки;
- управление контентом – администратор сайта в праве изменять или добавлять текстовый контент на сайте;
- проверка статуса заявки – статус заявки проверяется администратором (выполнена или не выполнена).

На основе DFD-декомпозиции бизнес-процесса управления заявками «как должно быть» создается диаграмма вариантов использования.

Дополнительная информация об отношениях между различными вариантами использования и внешними пользователями (актерами) предоставляется диаграммой вариантов использования.

2.1.2 Диаграмма классов

Диаграмма классов предназначена для изображения статической структуры модели системы в терминологии классов объектно-ориентированного программирования.

Диаграмма классов отражает, различные отношения между некоторыми сущностями предметной области, такими как подсистемы и объекты, а также изображает их типы отношений и внутреннюю структуру. На диаграмме классов не указываются данные о временных аспектах функционирования информационной системы. Следовательно, данная диаграмма является следующим развитием концептуальной модели проектируемой информационной системы.

Класс в языке UML служит для обозначения множества объектов, которые обладают одинаковой структурой, поведением и отношениями с объектами из других классов.

На рисунке 2.3 изображена диаграмма классов информационной системы.

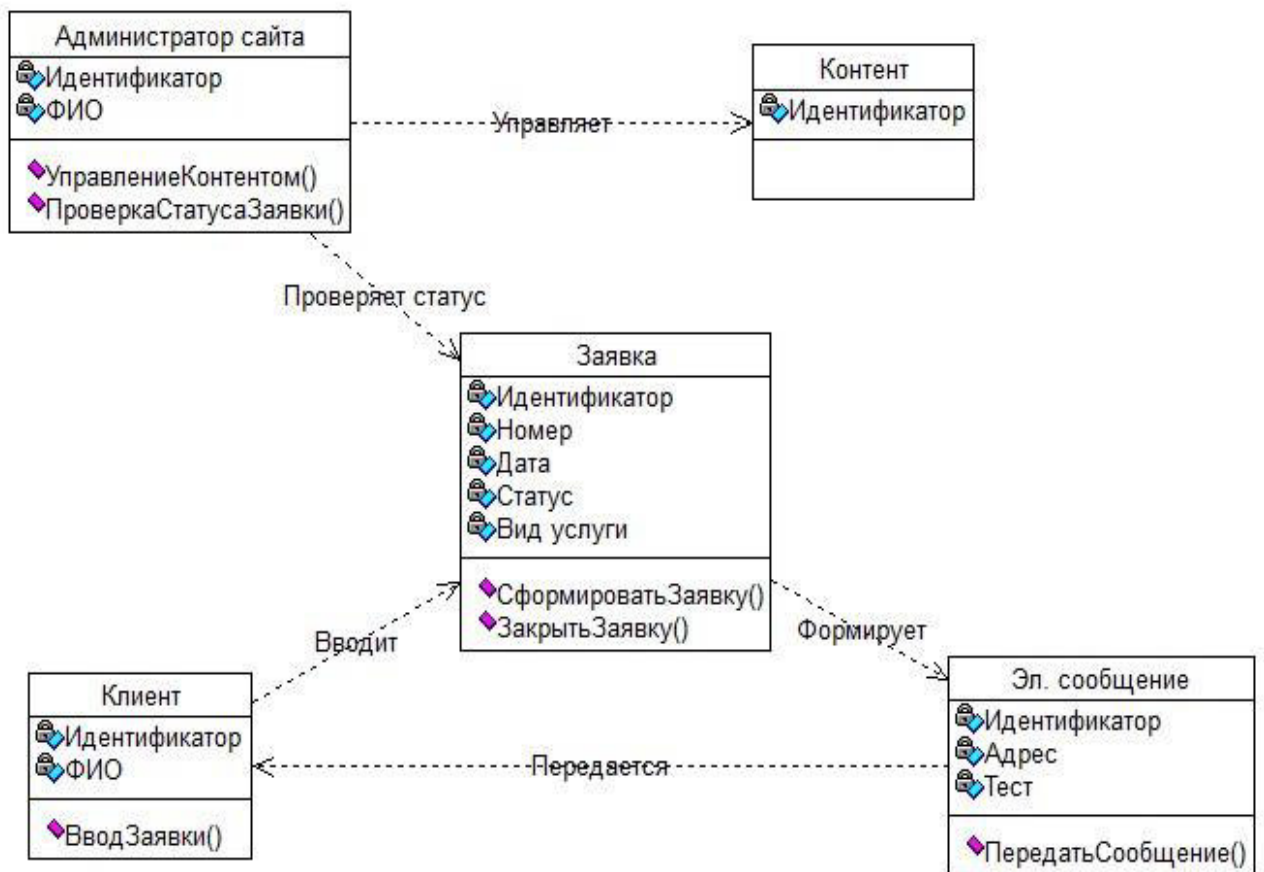


Рисунок 2.3 – Диаграмма классов ИС

Спецификация классов ИС:

- клиент – класс объектов-пользователей информационной системы, которые выполняют ввод заявок онлайн;
- администратор сайта – класс объектов-пользователей ИС, управляющих контентом сайта;
- заявка – класс объектов – электронных заявок;
- электронное сообщение – это класс объектов-почтовых сообщений;
- контент – класс объектов – текстовых содержаний ИС.

Диаграмма классов предназначена для разработки объектной модели приложения информационной системы.

2.1.3 Диаграмма последовательности

При моделировании взаимодействия предметов в языке UML применяются диаграммы взаимодействия.

Диаграмма последовательности используется для представления временных особенностей передачи и приема сообщений между объектами, когда взаимодействия объектов можно рассматривать во времени. Только те объекты, которые непосредственно участвуют во взаимодействии и не показываются возможные статические ассоциации с другими объектами могут изображаться на диаграмме последовательности.

Именно динамика взаимодействия объектов во времени является ключевым моментом для диаграммы последовательности.

Диаграмма последовательности управления контентом информационной системы приведена на рисунке 2.4.



Рисунок 2.4 – Диаграмма последовательности управления контентом информационной системы

В некоторый момент времени объект «администратор сайта» посылает объекту «контент» сообщение «изменить контент».

Объект «контент» изменяет текст контента, например, блока новостей ИС.

Процесс завершается.

Диаграмма последовательности ввода заявки представлена на рисунке 2.5.

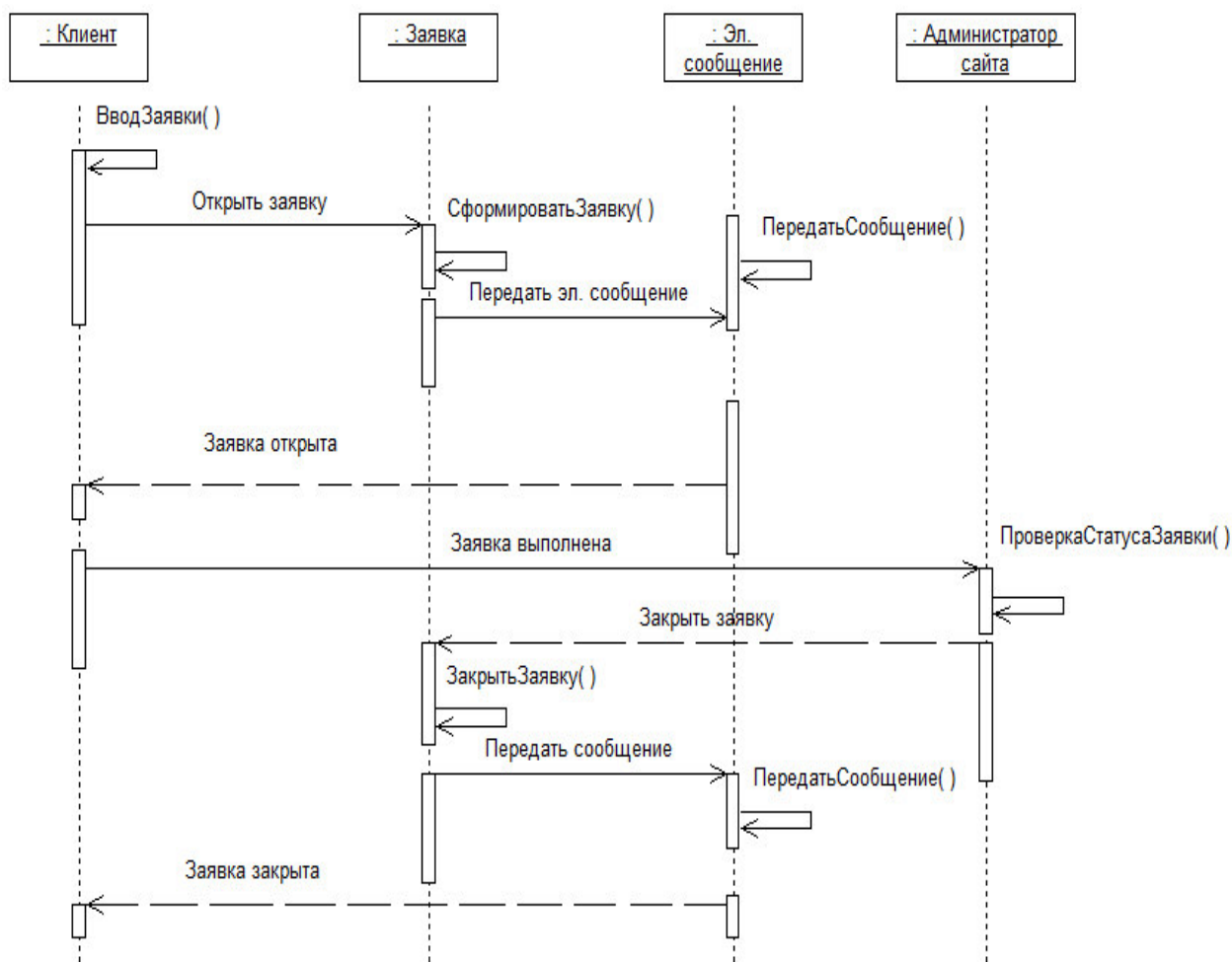


Рисунок 2.5 - Диаграмма последовательности ввода заявки

В какой-то момент времени объект «клиент» вводит онлайн-заявку и отправляет объекту «заявка» сообщение «открыть заявку».

Объект «заявка» формирует заявку и посылает объекту «электронное сообщение» команду «передать сообщение» объекту «клиент».

Объект «электронное сообщение» посылает по электронной почте сообщение объекту «клиент» об открытии заявки.

Объект «клиент» в случае выполнения заявки отправляет объекту «администратор сайта» сообщение «заявка выполнена».

Объект «администратор сайта» проверяет статус заявки и в случае его подтверждения посылает объекту «заявка» сообщение «закрыть заявку».

Объект «заявка» закрывает заявку и посылает объекту «электронное сообщение» команду «передать сообщение» объекту «клиент».

Объект «электронное сообщение» посылает по электронной почте письмо объекту «клиент» о закрытии заявки.

Процесс завершается.

2.1.4 Разработка логической модели данных

Процедура логического моделирования данных информационной системы является выражением интеграции и взаимосвязи реляционной и объектной технологий.

На сегодняшний день реляционные системы управления базами данных (СУБД) являются главным способом создания баз данных и хранилищ данных. Реляционная технология применяется достаточно давно и освоена большим количеством разработчиков и пользователей[2].

Реляционная модель довольно элементарна и обладает строгим математическим основанием, имеется большое разнообразие промышленных способов реализации, эксплуатации и проектирования реляционных баз данных.

Будет использована методология IDEF1X для проектирования реляционных баз данных[4].

В методологии IDEF1X модель данных представляется на двух уровнях: физическом и логическом.

Логическая модель данных – это независимое логическое представление концептуальной модели данных ИС.

Физическая модель данных содержит определения всех реализуемых объектов в конкретной базе данных и для конкретной СУБД.

Диаграмма классов информационной системы КСК превращается в ER-модель (диаграмму «сущность-связь»), в которой имеются новые сущности, так необходимые для обеспечения уровня нормализации данных. Это делается для создания логической модели данных.

Следующие основные сущности выделены в логической модели данных:

- клиент;
- администратор;
- заявка;
- электронное сообщение.

Следующие связи установлены между сущностями:

- клиент вводит заявку («один ко многим»);
- администратор управляет доступом клиента к ресурсу через сущность «аккаунт» («один к одному»);
- электронное сообщение формируется заявкой («один к одному»).

На рисунке 2.6 изображена логическая модель данных ИС КСК, которая обеспечивает поддержку модуля управления заявками жильцов, созданная на базе методологии IDEF1X.

Сущности статус, адрес и услуга выполняют справочные функции и обеспечивают нужную величину нормализации данных.

При проектировании физической модели реляционной БД системы в качестве основы будет использована представленная логическая модель, являющаяся концептуальной моделью данных ИС КСК.

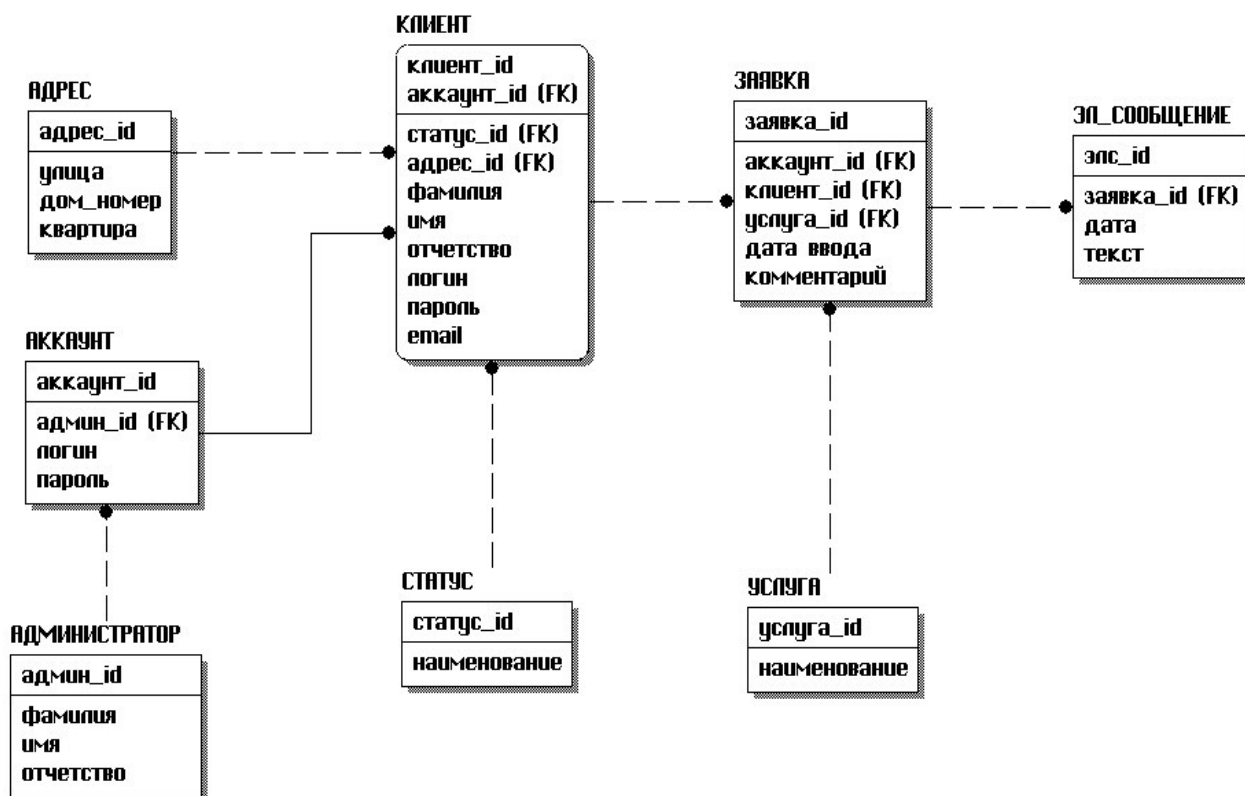


Рисунок 2.6 – Логическая модель данных ИС КСК

2.2 Возможное поле разработки

2.2.1 Выбор архитектуры

Современные веб-сайты создаются с помощью архитектуры «клиент-сервер».

Архитектура «клиент-сервер» может быть показана в следующих двух вариантах:

- двухзвенная архитектура «клиент-сервер». На рисунке 2.7 изображена модель двухзвенной архитектуры.

Первое звено двухзвенной архитектуры – это клиентские компьютеры с прикладными программами, используя которые пользователи могут обратиться к базе данных по сети («толстый клиент»).

Второе звено двухзвенной архитектуры – это сервер баз данных (СБД), который непосредственно участвует в обработке данных.

При использовании двухзвенной архитектуры снижается нагрузка на информационную сеть, по той причине, что отправляются только запросы и ответы на них.

Преимущества:

- сравнительно несложная реализации;
- сравнительно малая цена на серверное оборудование.

Недостатки:

- применение устаревших IT-решений;
- относительно высокая цена на оборудование клиентов.

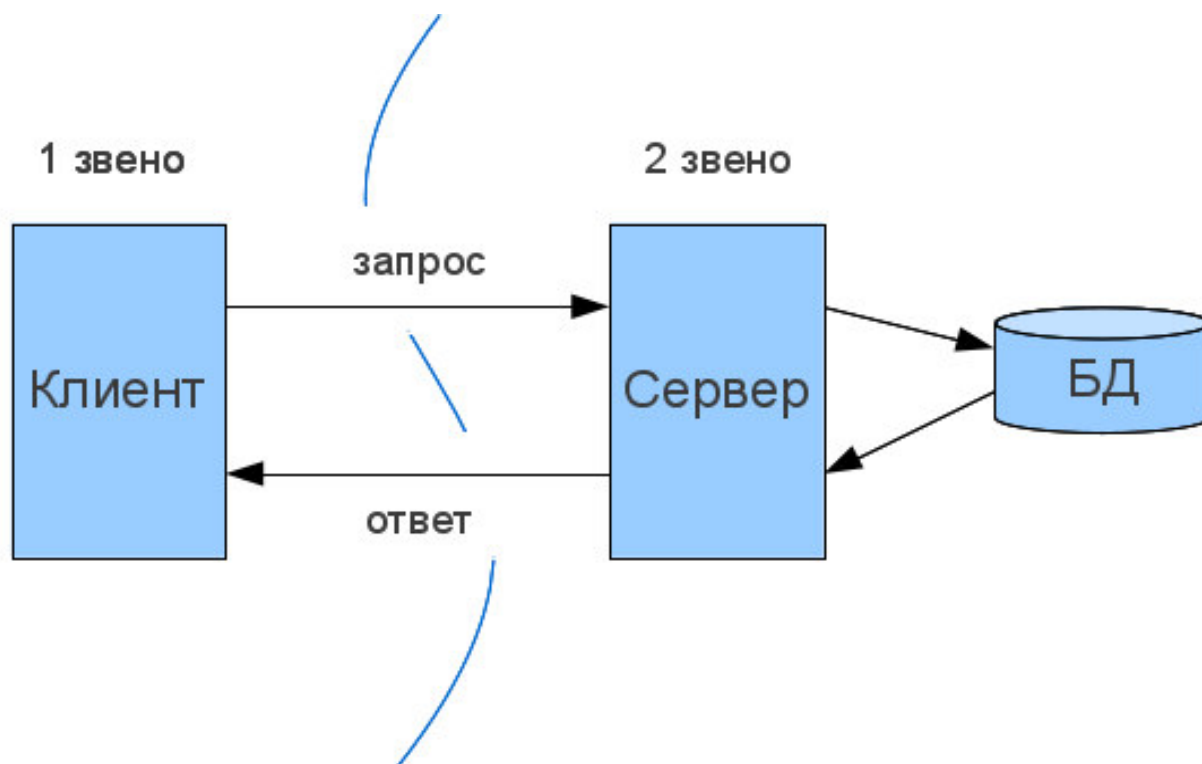


Рисунок 2.7 – Модель двухзвенной архитектуры

– трехзвенная архитектура «клиент-сервер». На рисунке 2.8 изображена модель трехзвенной архитектуры.

Первое звено трехзвенной архитектуры – веб-браузер (Internet Explorer, Opera, Chrome и др.) – «тонкий клиент».

Второе звено трехзвенной архитектуры – сервер приложений – ПО промежуточного уровня. Он синхронизирует работу компонентов системы и создает связи между ними. Как правило такую задачу ставят перед собой сетевые службы и приложения, в их числе веб-серверы.

Третье звено трехзвенной архитектуры – сервер баз данных. Данный сервер не работает напрямую с клиентскими программами. Такие действия повышают безопасность информации в системе.

Преимущества:

- масштабируемость;
- конфигурируемость – обособленность уровней друг от друга позволяет быстро и простыми средствами переконфигурировать систему при появлении ошибок или при плановом обслуживании на уровнях.

Недостатки:

- создание таких приложений является относительно сложной задачей;
- высокая цена на серверное оборудование, так как требования к производительности серверов приложений и сервера базы данных высокие.

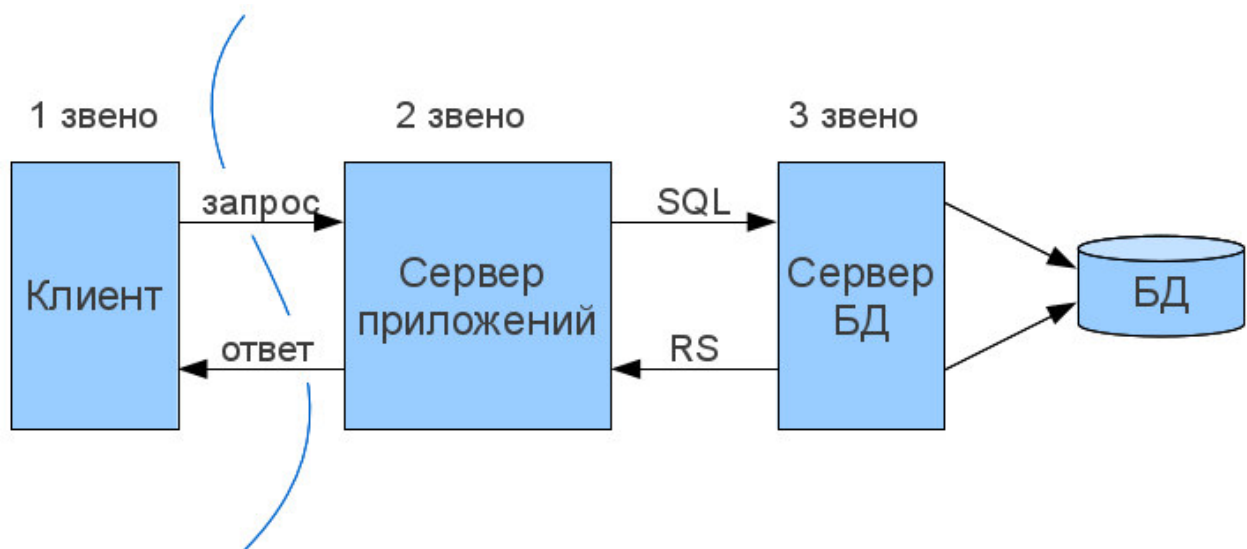


Рисунок 2.8 – Модель трехзвенной архитектуры

Сравним результаты проведенного анализа архитектур реализации ИС КСК в табл. 2.1.

Таблица 2.1 – Сравнительный анализ архитектур ИС КСК

Характеристика	Клиент-сервер, двухзвенная модель	Клиент-сервер, трехзвенная модель
Масштабируемость	-	+
Конфигурируемость	-	+
Маленькие затраты на эксплуатацию и разработку	+	-
Малые требования к оборудованию	+	-
Поддержка современных веб-технологий	-	+
Баллы	2	3

По результатам сравнения предлагается использовать трехзвенную архитектуру «клиент-сервер» при создании ИС КСК.

2.2.2 Выбор сервера баз данных

Следует учитывать требование обеспечить низкую стоимость затрат при выборе СУБД, которая будет использована в качестве СБД.

Произведем сравнительный анализ бесплатно распространяемых СУБД PostgreSQL, MySQL, FireBird и SQLite.

PostgreSQL не просто реляционная, а объектно-реляционная СУБД. Это даёт ему некоторые преимущества над остальными SQL базами данных с открытым исходным кодом, например, MySQL и FireBird. Фундаментальная характеристика объектно-реляционной БД – это поддержка пользовательских объектов и их действия, включая операции, домены, функции, индексы и типы данных. Благодаря таким характеристикам PostgreSQL является гибким и надежным. Помимо этого, он способен извлекать, хранить и создавать сложные структуры данных.

Существует обширный список типов данных, которые поддерживает PostgreSQL. Помимо поддержки текстовых, булевы, числовых и других ожидаемых типов данных и множество их вариаций, PostgreSQL отличается поддержкой uuid, геометрического, перечисляемого, бинарного и денежного типов, битовых строк, текстового поиска, сетевых адресов, json, xml, массивов, композитных типов и диапазонов, а также некоторых внутренних типов для идентификации объектов и расположения логов. MySQL и FireBird также обладают некоторыми из этих типов данных, но только PostgreSQL поддерживает их все.

PostgreSQL обеспечивает хранение разных типов сетевых адресов. Тип данных CIDR (бесклассовая маршрутизация интернет домена, Classless Internet Domain Routing) следует соглашению для сетевых адресов IPv4 и IPv6.

Также для хранения сетевых адресов доступен тип данных INET, используемый для IPv4 и IPv6 хостов, где подсети являются необязательными. Тип данных MACADDR может использоваться для хранения MAC-адресов для идентификации оборудования.

У MySQL тоже есть INET функции для конвертации сетевых адресов, но она не предоставляет типы данных для внутреннего хранения сетевых адресов. У FireBird тоже нет типов для хранения сетевых адресов.

Поскольку PostgreSQL – это объектно-реляционная база данных, массивы значений способны храниться для множества существующих типов данных чего не могут MySQL и FireBird. Для того, чтобы хранить такие массивы значений в традиционных реляционных базах данных, придется использовать обходной путь и создавать отдельную таблицу со строками для каждого из значений массива. Геоданные быстро становятся основным требованием для многих приложений. PostgreSQL уже давно поддерживает множество геометрических типов данных, таких как точки, линии, круги и многоугольники. Один из этих типов – PATH, он состоит из множества последовательно расположенных точек и может быть открытым (начальная и конечная точки не связаны) или закрытым (начальная и конечная точки связаны).

Поддержка JSON в PostgreSQL позволяет вам перейти к хранению schemaless данных в базе данных. Что может быть довольно полезным, в случаях, когда структуре данных необходима некоторой гибкости: к примеру, если в процессе создания структура до сих пор меняется или неизвестно, какие поля будет иметь объект данных. Тип данных JSON гарантирует

проверку корректности JSON, позволяющий использовать специализированные функции и операторы, которые встраиваются в PostgreSQL для реализации запросов и манипулирования данными. Еще доступен тип JSONB – двоичная разновидность формата JSON, у которой удаляются пробелы, а сортировка объектов не сохраняется, ведь вместо этого они хранятся наиболее оптимальным образом, в результате чего сохраняется только конечное значение для ключей-дубликатов. JSONB как правило является более предпочтительным форматом, так как требует меньше места для объектов, обрабатывается быстрее, так как не требует повторного синтаксического анализа и может быть проиндексирован.

PostgreSQL может обрабатывать много данных. Текущие опубликованные ограничения перечислены ниже:

Таблица 2.2 – Таблица максимальных ограничений по размеру для PostgreSQL

Наименьший размер базы данных	Неограничен
Наименьший размер таблицы	31,9 TB
Наименьший размер строки	1.59 TB
Наименьший размер поля	999 MB
Наименьшее количество строк в таблице	Неограниченно
Наименьшее количество столбцов в таблице	260-1500 в зависимости от типа столбца
Наименьшее количество индексов в таблице	Неограниченно

Уменьшение скорости работы команд манипулирования данными при больших массивах информации является недостатком PostgreSQL.

MySQL – это одна из самых популярных и самых распространенных СУБД в интернете. MySQL имеет высокое быстродействие благодаря внутреннему механизму многопоточности.

Благодаря создающейся при установке пакета и содержащей пять таблиц базе данных MySQL, обеспечивается высокий уровень безопасности. Можно описать, какой именно пользователь из какого домена с какой таблицей может работать и какие команды он в праве использовать, если воспользоваться этими таблицами. В MySQL встроена функция password(), благодаря которой можно зашифровать пароли, хранящиеся в базе данных.

Немаловажным фактором является ее бесплатность. MySQL распространяется на условиях общей лицензии GNU (GPL, GNU Public License). Благодаря чему возможно добавление нужных функций в пакет, увеличивая его функциональность так, как это требуется.

Надёжность и стабильность СУБД находится на очень высоком уровне, имеются многочисленные инструменты для восстановления таблиц (на тот случай если интенсивно работающий сервер внезапно обесточивается и т.п.) и

для увеличения надёжности (бинарные журналы, репликация, автоподсчёт контрольной суммы и т.п.).

MySQL не требует высокопроизводительного компьютера для быстрой работы.

Недостатком MySQL является то, что она не предназначена для работы с большими объемами данных, но её применение идеально для интернет-сайтов, как небольших, так и достаточно крупных.

FireBird – компактная реляционная, кроссплатформенная СУБД. Может работать под управлением многих операционных систем - Windows, MacOS, Linux, Solaris и различных Unix-платформ, а также полностью поддерживает стандарты ANSI в синтаксисе языка SQL. К достоинствам этой системы можно отнести использование очень развитого языка для хранимых процедур и триггеров.

В качестве достоинств FireBird можно также добавить многоверсионную архитектуру, которая обеспечивает параллельную обработку аналитических и оперативных запросов, которые возможны потому, что читающие пользователи не блокируют пишущих, компактность с дистрибутивом в 5Mb, мощную языковую поддержку для хранимых процедур и триггеров и высокую эффективность. С 2001 года FireBird используется в различных промышленных системах, в финансовых, складских, хозяйственных и государственных секторах.

Так как FireBird является сервером баз данных, один его сервер может обрабатывать несколько сотен независимых баз данных, каждая из которых может иметь множество пользовательских соединений. FireBird полностью свободная от лицензионных отчислений база данных, в том числе и для коммерческого использования.

Транзакции находятся под полным контролем: множество одновременных транзакций могут быть выполнены одним клиентским приложением. Разные уровни изоляции могут быть использованы в разных транзакциях. Гарантированную устойчивость при работе с несколькими базами данных обеспечивает протокол двухфазного подтверждения транзакций. Так же доступны точки сохранения транзакций и оптимистическое блокирование данных.

Нет необходимости останавливать сервер при выполнении резервного копирования. Вы можете работать с базой данных при выполнении процесса резервного копирования и FireBird все равно сохранит состояние базы данных на момент своего старта. Помимо этого, существует возможность совершать инкрементальное резервное копирование базы данных.

Можно назначить несколько триггеров, срабатывающих до или после ставки, обновления или удаления записей для каждой таблицы. Для триггеров применяется язык PSQL, это позволяет вносить начальные значения, проверять целостность данных, вызывать исключения, и т.д. Начиная с версии 1.5 в FireBird появились «универсальные» триггеры, позволяющие в

одном триггере обрабатывать вставки, обновления и удаления записей таблицы.

Библиотеки с UDF (User Defined Function) могут быть написаны на любом языке, и легко подключаться к серверу в виде DLL/SO, что позволяет расширять возможности сервера «изнутри».

Отсутствие кэша полнотекстовых индексов, результатов запросов, значительное падение производительности при росте внутренней фрагментации базы – основные недостатки FireBird.

SQLite – база данных, способная легко встраиваться в приложения. Она предоставляет довольно большой набор инструментов для работы с ней, по сравнению с сетевыми СУБД так как это система базируется на файлах. Обращения происходят напрямую к файлам, хранящим данные, вместо портов и сокетов в сетевых СУБД. По этой причине SQLite считается очень быстрой, а также мощной, благодаря технологиям обслуживающих библиотек, базой данных.

Типы данных SQLite:

- Text – текстовая строка, использующая кодировки UTF-8, UTF-16BE или UTF-16LE при хранении;

- Integer – знаковое целочисленное значение, которое в зависимости от порядка числа использует 1, 2, 3, 4, 6, или 8 байт;

- Real – это число с плавающей точкой, для хранения числа в формате IEEE занимает ровно 8 байт;

- NULL – значение NULL;

- Blob – тип данных blob, массив двоичных данных. В первую очередь предназначен для хранения изображений, аудио и видео.

Преимущества SQLite:

- файловая структура - вся база данных состоит из одного файла, поэтому её очень легко переносить на разные машины;

- используемые стандарты - хотя может показаться, что эта СУБД примитивная, но она использует SQL. Некоторые особенности опущены (right outer join или for each statement), но основные все-таки поддерживаются;

- отличная при разработке и тестировании - в процессе разработки приложений часто появляется необходимость масштабирования. SQLite предлагает всё что необходимо для этих целей, так как состоит всего из одного файла и библиотеки написанной на языке C.

SQLite имеет следующие недостатки:

- отсутствие системы пользователей – более крупные СУБД включают в свой состав системы управления правами доступа пользователей. Обычно применения этой функции не так критично, так как эта СУБД используется в небольших приложениях;

- отсутствие возможности увеличения производительности – исходя из проектирования, довольно сложно получить что-то более производительное из этой СУБД.

SQLite можно использовать как встроенные приложения. Если необходимо легко перенести приложения, а масштабируемость не важна. Например, мобильные приложения, игры или однопользовательские приложения. Программу можно использовать как прямой доступ к диску или при тестировании. Вы можете выиграть при переходе на эту СУБД в функционале и простоте использования SQL языка при необходимости напрямую обращаться к диску. Но приложение сильно замедляется при использовании дополнительных процессов при тестировании функционала.

SQLite не стоит применять как многопользовательские приложения, при необходимости обеспечить доступ к данным для нескольких пользователей, а также различать их по правам доступа, то полноценная будет более логичным выбором. Запись огромных объемов данных – это одно из ограничений SQLite. Производительность сильно ограничивается по той причине, что разрешен только один процесс записи в промежуток времени.

Результаты сравнительного анализа СУБД для информационной системы КСК приведены в таблице 2.3.

Таблица 2.3 – Сравнительный анализ СУБД

Характеристики	PostGreSQL	MySQL	FireBird	SQLite
Быстродействие	-	+	-	+
Поддержка транзакций	+	+	+	+
Поддержка реляционных баз данных	+	+	+	+
Функция масштабирования	-	-	-	+
Баллы	2	3	2	4

По результатам сравнительной таблицы в качестве базы данных ИС КСК будет использована СУБД SQLite.

2.2.3 Выбор технологии программирования

Важно учесть следующие критерии при подходе к разработке программного обеспечения в рамках спиральной модели его жизненного цикла при выборе средств разработки:

- поддержка технологии скоростной разработки приложений RAD (Rapid Application Development), основанной на объектно-ориентированной парадигме программирования;
- полная поддержка трехзвенной архитектуры «клиент-сервер»;
- разработчиком должен знать язык;

- поддержка базы данных SQLite;
- низкая стоимость на разработку ПО.

На сегодняшний день для разработки динамических сайтов часто используются такие технологии веб-программирования, как ASP.NET, Java, PHP и Python.

ASP.NET – технология разработки веб-приложений компании Microsoft. Эта технология является частью платформы Microsoft.NET. В целях достижения наибольшей производительности рекомендуется использовать данную технологию вместе с остальными продуктами компании Microsoft.

Платформа не является бесплатной, но представляет собой более строгую систему, чем PHP и поэтому считается, что программирование динамического сайта на ASP.Net проще для разработчиков.

В качестве главного недостатка ASP.Net обычно называют меньшую производительность сайтов, чем ресурсов на MySQL и PHP.

В тех случаях, когда системе управления сайтом приходится обрабатывать в одновременном процессе огромное число запросов – ASP.Net уверенно выходит вперед. Поэтому можно наблюдать платформу ASP.Net на сайтах банков и других клиент-ориентированных ресурсов.

Кроме всего прочего, соотношение количества опытных программистов, умеющих работать с ASP.Net примерно в полтора раза меньше, чем для PHP. Еще одно замечание, касающееся разработки сайтов на основе ASP.Net – наилучшие результаты достигаются в командной работе, когда каждый программист является спецом в одном аспекте ASP.Net и выполняет узкие задачи.

Все это навеивает мысли, что создание сайтов на платформе ASP.Net больше подходит для масштабных проектов, имеющих достаточные финансовые ресурсы для содержания или аутсорсинга команды подготовленных ASP.Net-программистов.

Еще одно существенное уточнение. Поскольку ASP.Net является проприетарной платформой с закрытым кодом, то очевидно, что степень защищенности от киберпреступности здесь будет выше, чем для свободно распространяемых, общедоступных PHP-разработок.

Поэтому ASP.Net целесообразно применять для создания проектов, где требуется повышенная степень защищенность.

Java – это очень популярный объектно-ориентированный язык программирования. Разработан компанией Sun, а затем приобретен компанией Oracle.

Java включает в себя объектно-ориентированное программирование – концепцию, в которой вы не только определяете тип данных и его структуру, но и набор функций, применяемых к нему. Таким образом, структура данных становится объектом, которым можно управлять для создания отношений между различными объектами.

Java – это язык высокого уровня, то есть он похож на человеческий язык. В отличие от языков низкого уровня, которые напоминают машинный

код. Языки высокого уровня преобразуется с помощью компиляторов или интерпретаторов. Это упрощает разработку, делая язык более легким для написания, чтения и обслуживания.

Синтаксис Java основан на C ++, поэтому Java похожа на C. Тем не менее, синтаксис Java проще, что позволяет новичкам быстрее учиться и эффективнее использовать код для достижения конкретных результатов.

Корпоративные приложения - главное преимущество Java с 90-х годов, когда организации начали искать надежные инструменты программирования не на C. Java поддерживает множество библиотек- строительных блоков любой корпоративной системы. Библиотеки помогают разработчикам создавать любые функции, которые могут понадобиться компании. Java широко распространен, это язык, который преподают в рамках введения в программирование в большинстве школ и университетов. Возможности интеграции Java впечатляют: большинство хостинг-провайдеров поддерживают Java. Более того, Java-язык, дешевый в обслуживании: работать с Java можно с любого компьютера, вне зависимости от конкретной аппаратной инфраструктуры.

Существует мнение, что Java – безопасный язык, однако это не совсем так. Сам язык не защищает вас от уязвимостей, но некоторые его функции устраняют распространенные уязвимости. Во-первых, в отличие от C, в Java нет указателей. Указатель - это объект, который сохраняет адрес ячейки памяти другого значения, что может вызвать несанкционированный доступ к памяти. Во-вторых, в Java есть Security Manager, созданная для каждого приложения политика безопасности, в которой можно указать правила доступа. Это позволяет запускать приложения в «песочнице», устраняя уязвимости.

Достоинства Java:

- среди всех языков программирования имеет наибольшую степень переносимости программ;
- имеет мощные стандартные библиотеки;
- имеет встроенную поддержку работы как в локальных сетях, так и в сетях Internet/Intranet.

Недостатки Java:

- если сравнивать с другими языками, то Java имеет повышенные требования к объему оперативной памяти и низкое быстродействие;
- сложности в изучении языка связаны с большим объемом стандартных библиотек и технологий;
- наличие как устаревших, так и новых средств, имеющих одно и то же функциональное назначение вызвано постоянным развитием языка.

PHP – это интенсивно применяемый для разработки динамических веб-приложений скриптовый язык общего назначения. На сегодняшний день является одним из лидеров среди языков, применяющихся для создания

динамических веб-приложений и поддерживается большим количеством хостинг-провайдеров.

Задачей языка PHP является обеспечение эффективной связи веб-ресурса с сервером и базами данных. При этом данный язык необычайно прост в освоении и применении. По сути, все, что необходимо знать для начала изучения php и программирования на нем, это самый простой язык веб-разметки HTML. PHP прекрасно сочетается с HTML-кодом. Однако для выполнения сценариев одного лишь браузера недостаточно. Необходим веб-сервер.

Код, написанный на PHP, направлен на выполнение двух задач:

- html-часть отвечает за внешний вид и отображение информации;
- php-часть, интегрированная в html, обеспечивает возможности интерактивности и динамику.

При этом подобные программные коды и, соответственно, составленные с их помощью проекты, являются легкими, эффективными, гибкими, многофункциональными, удобными в администрировании, редактировании, обслуживании.

На сегодняшний день подавляющее большинство сайтов, сервисов и приложений, а также такие популярные платформы как Joomla, Drupal, WordPress, 1С-Bitrix и UMI.CMS написаны именно на PHP-языке.

Язык PHP обладает рядом неоспоримых преимуществ:

- высокая скорость работы и, соответственно, общая производительность ресурсов;

- бюджетность, экономичность. Найти специалиста не представляется проблемой, стоимость написания программ на php не высока;

- простота освоения, простой синтаксис;

- отличная совместимость и переносимость – php-коды работают одинаково хорошо с разными платформами;

- набор текста кода и его редактирование можно осуществлять в любом текстовом или html-редакторе;

- высокая гибкость, емкость и функциональность. PHP-программу можно составлять отдельно от разработки веб-страницы, без привязки, после чего совместить. Это существенно упрощает жизнь дизайнеров и программистов;

- многозадачность и широкие возможности – создание любых веб-приложений, блогов, гостевых книг, интернет-магазинов, сайтов, работа с редиректами, заголовками, pdf-документами, базами данных, электронной почтой и пр.

Python – это интерпретируемый, изначально объектно-ориентированный языком программирования. Он не только выразителен и гибок, но и чрезвычайно прост и содержит небольшое число ключевых слов. За счет встроженных высокоуровневых структур данных (списки, тьюплы, словари)

Python находится на более высоком уровне нежели Pascal, C++ и, естественно C.

К его достоинствам можно отнести то, что интерпретатор Python работает практически на всех операционных системах и платформах. Язык C был первым таким языком, но его типы данных на различных машинах могли занимать различный объем памяти, что служило препятствием при написании действительно переносимой программы. И в отличие от языка C, Python не обладает таким недостатком.

Следующая важная особенность – это расширяемость языка, то есть любой заинтересованный программист при желании имеет возможность усовершенствовать язык. Его исходный код доступен для любых манипуляций, а интерпретатор написан на языке C. Можно вставить его в свою программу и использовать как встроенную оболочку в случае необходимости. Имеется возможность получить расширенный интерпретатор с новыми возможностями написав на языке C свои дополнения к Python и скомпилировав программу.

Следующим достоинством Python является наличие значительного числа подключаемых к программе модулей, которые способны обеспечивать различные дополнительные возможности. Любые достаточно квалифицированные программисты имеют возможность разработать такие модули, написав их на языке C и на самом Python. Например, имеются следующие модули:

- OpenGL – применение широкой библиотеки графического моделирования двух и трехмерных объектов Open Graphics Library фирмы Silicon Graphics Inc. Данный стандарт также поддерживается, в таких распространенных операционных системах как Microsoft Windows 95 OSR 2, 98 и Windows NT 4.0;

- Tkinter – создание приложений с применением графического пользовательского интерфейса (GUI) на базе широко распространенного на X-Windows Tk-интерфейса;

- Numerical Python – обширные математические возможности, например, манипуляции с целыми матрицами и векторами.

Единственным недостатком, является относительно медленная скорость выполнения Python-программы, напрямую связанное с ее интерпретируемостью. Однако этот недостаток с лихвой компенсируется достоинствами языка при написании программ, не сильно зависящих от скорости ее выполнения.

Python, в отличие от таких языков как Pascal, C++, Java и других, не нуждается в описании переменных. При первом присваивании переменной какого-либо значения эти переменные создаются в месте их инициализации. Это значит, что тип присваиваемого значения определяет тип переменной. В этом смысле Python напоминает язык Basic. Тип переменной никогда не является неизменным. Какое бы то ни было присваивание для переменной

будет корректным, и это приводит к тому, что тип нового присваиваемого значения станет типом переменной.

Организация списков всегда представляла некоторые сложности в таких языках как Pascal, C, C++. Приходилось основательно осваивать принципы работы с указателями и динамической памятью для их реализации. Программист, каждый раз с нуля создавая механизмы создания, работы и уничтожения списков, мог легко допустить трудноуловимые ошибки даже имея хорошую квалификацию. По этой причине были разработаны некоторые средства для работы со списками. В Delphi Pascal это класс TList, который реализует списки; для C++ это библиотека STL (Standard Template Library), в которой содержатся такие структуры как очереди, словари, стеки, списки, множества и векторы. Однако, далеко не во всех языках и их реализациях имеются такие средства.

Важной отличительной чертой Python является присутствие таких встроенных в сам язык структур как списки(list), тьюплы(tuple) и словари(dictionary), которые иногда называют картами(map).

Python, как и Java, не поддерживает работу с указателями, динамической памятью и адресную арифметику в отличие от Pascal, C, C++. Для обеспечения большей надежности и простоты в Python не были включены указатели, которые служат источником трудноуловимых ошибок, а работа с ними скорее больше относится к программированию на довольно низком уровне.

То, как происходит присваивание одной переменной другой, т.е. когда по обе стороны от оператора "=" стоят переменные является одной из особенностей Python.

Результаты сравнительного анализа технологий веб-программирования приведены в таблице 2.4.

Таблица 2.4 – Сравнительный анализ технологий веб-программирования

Характеристики	ASP.NET	Java	PHP	Python
Объектно-ориентированный подход к разработке программного обеспечения	+	+	+	+
Трехзвенная архитектура «клиент-сервер»	+	+	+	+
Знание языка разработчиком	-	-	-	+
СУБД SQLite	+	+	+	+
Малая цена на разработку ПО	-	-	+	+
Баллы	2	3	4	5

В качестве основного средства разработки веб-сайта КСК выбрана технология веб-программирования Python.

2.2.4 Выбор веб-фреймворка

В сети доступно множество фреймворков для python, рассмотрим такие примеры как Django, Pyramid, Web2py, Flask, CherryPy, Bottle и Tornado.

Django - это самый популярный фреймворк, имеющий открытый исходный код. Он позволяет сделать разработку простой, код чистым, а освоение-быстрым, предоставляет функционал для масштабирования Python-проектов. Может добавить большинство стандартных функций единым пакетом вместо поиска отдельных библиотек. Среди них, такие востребованные, как аутентификация, URL-маршрутизация, миграция схемы данных и т.п.

Django использует ORM для сопоставления объектов с таблицами баз данных. Один и тот же код работает с разными базами данных, так что переход из одной БД в другую становится простой задачей. Основными базами, с которыми работает Django, являются PostgreSQL, MySQL, SQLite и Oracle. Можно добавить и другие, но для этого понадобятся решения сторонних разработчиков.

Django это универсальный фреймворк. Он может использоваться при создании любых сайтов, поскольку умеет доставлять веб-контент в различных форматах (например, JSON, XML, RSS). Также он дружелюбен к начинающим разработчикам.

Pyramid - фреймворк, который дает максимум возможностей разработчику с минимальными затратами времени и ресурсов.

Наиболее интересная возможность фреймворка - работа как с большими, так и с малыми приложениями. Отдельно стоит выделить такие функции Pyramid, как:

- однофайловые приложения;
- генерация URL;
- масштабируемая конфигурация;
- гибкая схема аутентификации и авторизации;
- доступная пониманию техническая документация.

Лучше всего он подходит для тех, кто занимается разработкой API, а также прототипированием и разработкой крупных веб-приложений.

Web2py - это масштабируемый фреймворк для Python. Он поставляется с собственным IDE which: в нем есть редактор кода, дебаггер и деплой в один клик. Среди прочих функций и инструментов стоит выделить следующие:

- отсутствие необходимости в установке и настройке;
- работа в среде Windows, Mac, Google App Engine, Amazon EC2 и на любом хостинге, который поддерживает Python 2.5–2.7 или Java+Python;

- работа с различными протоколами;
- высокий уровень безопасности данных;
- трекер ошибок;
- обратная совместимость, которая позволяет без труда работать с приложениями и сервисами на основе прежних версий фреймворка.

Flask - это микрофреймворк, и поэтому большее внимание в нем уделено простоте работы, а не функциям. Для настройки и установки нужно гораздо меньше времени, чем с другими фреймворками, а между вами и HTTP-функциями будет меньше уровней абстракции. Основная идея Flask заключается в том, чтобы помочь создать прочную основу веб-приложений.

Он имеет следующий ряд полезных функций: встроенный сервер разработки и быстрый отладчик, Jinja2 templating, поддержка безопасных файлов cookie, возможность подключать любую ORM.

Лучше всего подходит для веб-разработчиков, которые хотят использовать самые лучшие практики, добиться быстрого прототипирования и создавать автономные приложения.

CherryPy – это минималистичный веб-фреймворк с открытым исходным кодом. Он обеспечивает возможность создания веб-приложений Python, которое ничем не отличается от построения любой другой объектно-ориентированной программы. По факту это веб-приложение, работающее под управлением CherryPy, является автономным приложением Python, использующее собственный многопоточный веб-сервер.

Приложения CherryPy запускаются в любой операционной системе, поддерживающей Python (Windows, MacOS, Linux и т.д.). Они могут быть развернуты в любом окружении, где вы можете запускать обычное приложение Python. Вам не требуется Apache для приложений CherryPy, вы можете запустить приложение CherryPy в Apache, так же как в Lighttpd или IIS.

CherryPy позволяет использовать любой тип технологии для шаблонизации, доступа к данным и т.д. Он также может обрабатывать статьи, статику, cookie, загружать файлы и все остальное, что умеют другие фреймворки.

Его отличительные способности:

- простой запуск нескольких HTTP-серверов одновременно;
- мощная конфигурационная система;
- гибкая система плагинов;
- возможности «из коробки»: кэширование, декодирование, сессии, аутентификация, статический контент и т.п.;
- возможность работы под Python 2.7+, Python 3.1+, PyPy, Jython и Android.

Bottle – это микрофреймворк, который изначально предназначался для создания API. Имеет в наличии лишь только один единственный source-файл. Работа с ним не требует нестандартных решений, а его функции по

умолчанию включают в себя маршрутизацию, шаблоны, утилиты и абстракцию по стандарту WSGI:

- маршрутизация поддерживает запросы на отображение функциональных вызовов, позволяя вам получать чистые и динамические URL-адреса;

- шаблоны - много нужных инструментов, работающих «из коробки» вместе с полной поддержкой `marko`, `jinja2` и `cheetah`;

- утилиты - удобный доступ к данным, загрузка файлов, куки, заголовки и другие метаданные, связанные с HTTP;

- встроенный сервер разработки HTTP, который поддерживает функции `farws3`, `bjoern`, `GAE`, `CherryPy` и любой другой HTTP-сервер с WSGI.

Это отличное решение для прототипирования, изучения структуры веб-фреймворков и создания небольших приложений.

Tornado - это веб-фреймворк Python и асинхронная сетевая библиотека одновременно. Он использует неблокирующий сетевой ввод-вывод и решает проблему C10k (при правильной настройке он может обрабатывать 10000+ одновременных подключений).

Все это делает его отличным инструментом для создания приложений, требующих высокой производительности и работы с десятками тысяч пользователей одновременно.

Основные достоинства:

- встроенная поддержка аутентификации пользователей;

- услуги в режиме реального времени;

- высококачественная производительность;

- язык веб-шаблонов на основе Python;

- неблокирующий HTTP-клиент;

- внедрение сторонних схем аутентификации и авторизации (Google OpenID/OAuth, Facebook Login, Yahoo BBAuth, FriendFeed OpenID/OAuth, Twitter OAuth);

- поддержка переводов и локализаций.

Так как функционал приведённых ранее фреймворков несильно отличается друг от друга, то предпочтение отдается самому популярному фреймворку – Django.

2.2.5 Разработка физической модели данных

Физическая модель данных зависит от конкретной СУБД, фактически являясь отображением системного каталога.

В ней содержится информация о любых объектах БД.

На рисунке 2.9 изображена физическая модель базы данных ИС КСК, построенная для СУБД SQLite.

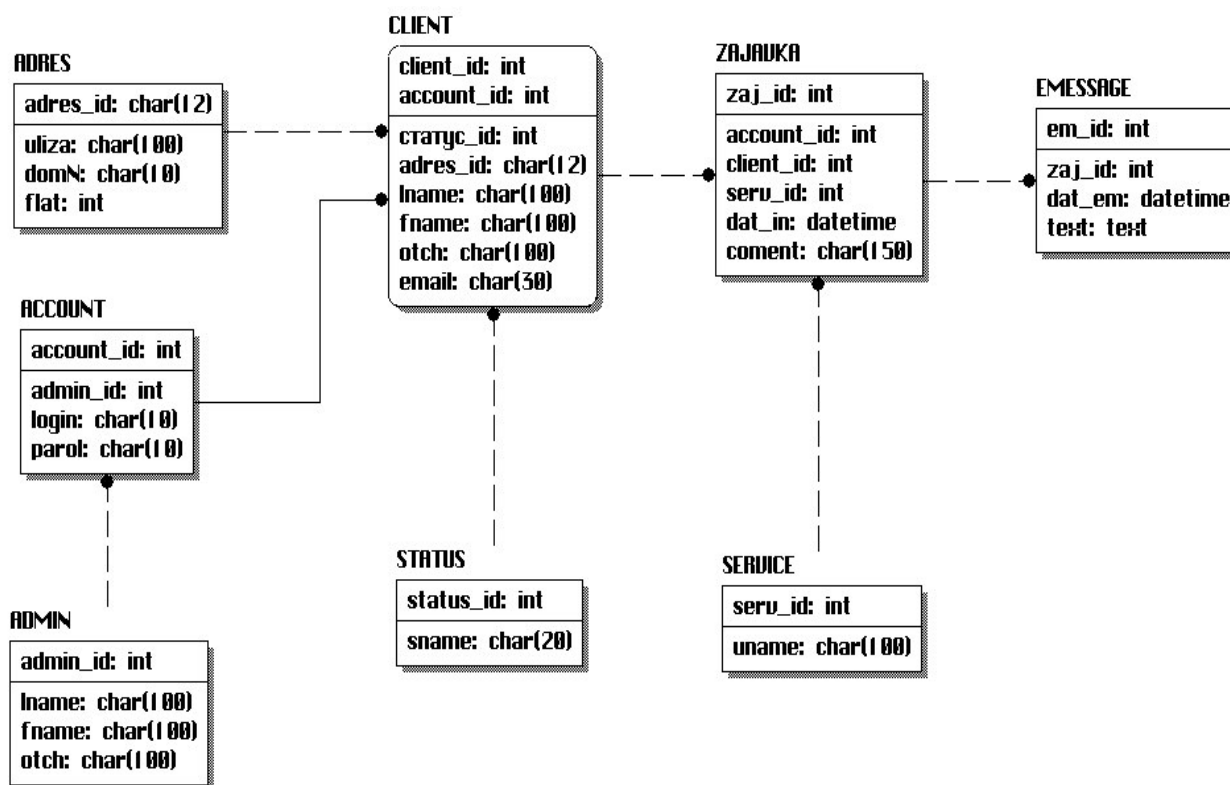


Рисунок 2.9 – Физическая модель базы данных ИС КСК

Непосредственно на основе логической модели данных системы разработана физическая модель данных.

2.2.6 Диаграмма размещения компонентов

Разрабатываемая информационная система состоит из следующих компонентов:

- клиент – веб-браузер (IE, Mozilla, Chrome);
- веб-фреймворк Django;
- сервер баз данных на основе системы управления базами данных SQLite.

Доступ к серверам сайта КСК будет осуществляться с помощью интернета.

Используем UML диаграмму размещения компонентов для представления взаимодействия компонентов сайта.

Диаграмма размещения компонентов веб-сайта КСК представлена на рисунке 2.10.

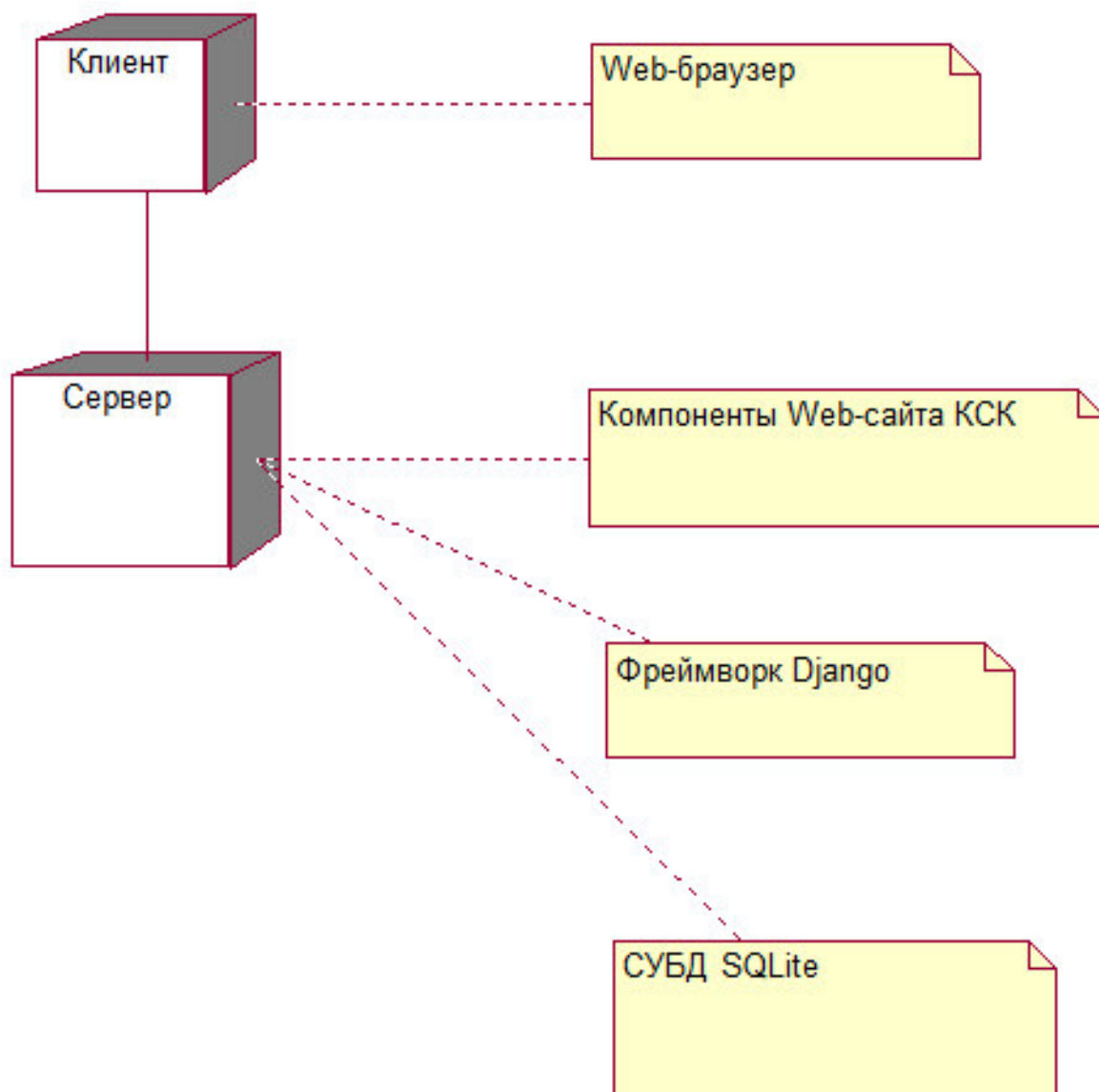


Рисунок 2.10 – Диаграмма размещения компонентов веб-сайта КСК

Диаграмма размещения компонентов отражает различные физические взаимосвязи между аппаратными и программными компонентами информационной системы. Она является отличным средством, когда необходимо показать размещение объектов и компонентов в распределенной информационной системе.

В приложении представлены фрагменты программного кода веб-сайта КСК.

2.2.7 Требования к аппаратно-программному обеспечению

Необходимо учесть следующие требования при выборе аппаратного обеспечения веб-сайта:

- обеспечить надежное функционирование выбранного программного обеспечения;
- иметь поддержку трехзвенной архитектуры «клиент-сервер»;

– обеспечить круглосуточную работу в любой день недели.

Имеющаяся IT-инфраструктура КСК «Батыр» не позволяет обеспечить решение описанных задач своими аппаратно-программными средствами, поэтому принято решение передать веб-сайт в хостинг.

3 Программная реализация

3.1 Главная страница

На главной странице размещена лента новостей, в левой части расположены ссылки на остальные страницы сайта. Вход в личный кабинет находится внизу левого блока страницы.

На рисунках 3.1 и 3.2 представлена главная страница сайта.

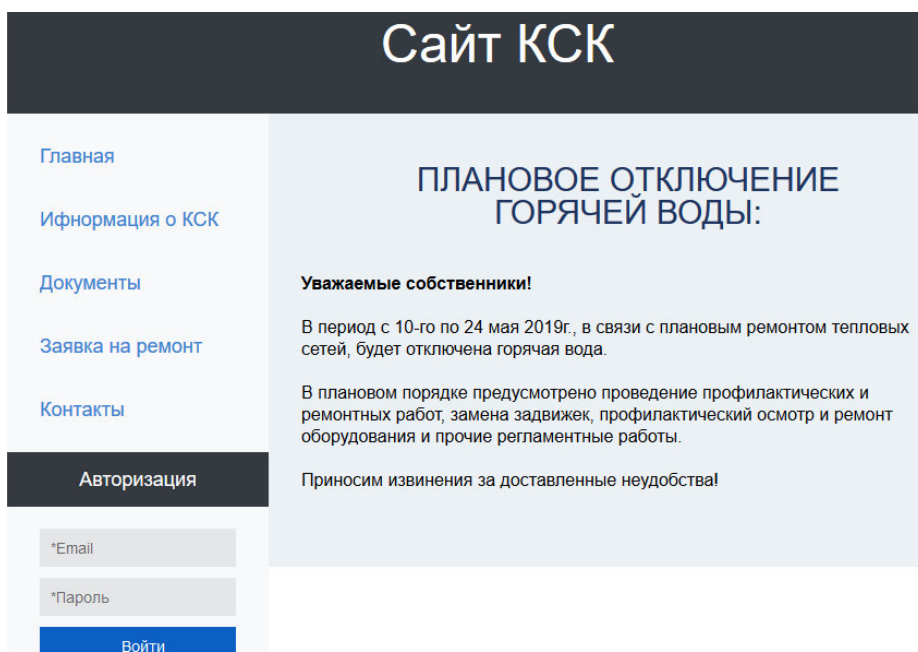


Рисунок 3.1 – Главная страница

На рисунке 3.2 можно увидеть фрагмент реализации главной страницы.

```
1  {% load staticfiles %}
2
3  <!DOCTYPE html>
4  <html lang='ru'>
5  <head>
6  <title>{% block title %}{% endblock %} - Сайт КСК</title>
7  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8  </head>
9  <body>
10
11  <div class="con">
12
13  <div class="header">
14  <h1>Сайт КСК</h1>
15  {% if auth_p == 'ok' %}
16  <a href="/signin/">Выйти</a>
17  {% if request.get_full_path != '/lich_kab/' %}
18  <a href="/lich_kab/">Личный кабинет</a>
19  {% endif %}
20  {% endif %}
21  </div>
22
23  <div class="left">
24
25  <ul>
26  <li><a href="/">Главная</a></li>
27  <li><a href="/info/">Информация о КСК</a></li>
28  <li><a href="/documents/">Документы</a></li>
29  <li><a href="/zayavka/">Заявка на ремонт</a></li>
30  <li><a href="/contacts/">Контакты</a></li>
31  </ul>
32
33  <div class="lich-kab">
34
35  {% if auth_p == 'no' %}
36  <div class="zag"><a>Авторизация</a></div>
37  <form action="/signin/" method="POST" class="fb" enctype="multipart/form-data">
```

Рисунок 3.2 – Код главной страницы

3.2 Регистрация

Жилец дома, обслуживаемый КСК, должен предварительно пройти регистрацию, в результате чего создается его учетная запись и личный кабинет.

Процесс регистрации является автоматизированным. Для того чтобы пройти регистрацию, помимо остальных личных данных, необходимо ввести номер договора купли-продажи квартиры, известный только жильцу и КСК. После этого происходит проверка номера по базе данных. В случае если такой номер имеется, пользователь успешно проходит регистрацию.

На рисунках 3.3 и 3.4 показана страница регистрации пользователя.

The screenshot shows a web interface for user registration. On the left, a sidebar menu includes 'Главная', 'Информация о КСК', 'Документы', 'Заявка на ремонт', 'Контакты', 'Авторизация' (highlighted), and 'Регистрация'. The main content area is titled 'РЕГИСТРАЦИЯ' and contains a form with the following fields: *Фамилия, *Имя, *Отчество, *Номер дома, *Квартира, *E-mail, *Пароль, *Подтверждение пароля, and *Договор Купли-Продажи. There is a 'Войти' button under the 'Авторизация' section and a 'Зарегистрироваться' button at the bottom of the registration form.

Рисунок 3.3 – Страница регистрации пользователя

```
1  {% extends 'article/base.html' %}
2  {% block title %}
3      Регистрация
4  {% endblock %}
5  {% block page_content %}
6      <h2>Регистрация</h2>
7
8      <form method="POST" class="fb">
9          {{ csrf_token }}
10         {{ form.last_name }}
11         {{ form.first_name }}
12         {{ form.patronymic }}
13         {{ form.home }}
14         {{ form.apartment }}
15         {{ form.email }}
16         {{ form.password }}
17         {{ form.password2 }}
18         {{ form.contract }}
19         {% if error.title %}<p style="color:red">{{ error.title }}</p>{% endif %}
20         <input type="submit" value="Зарегистрироваться" class="login">
21     </form>
22     <script>
23         var password = document.getElementById("password")
24         , confirm_password = document.getElementById("confirm_password");
25
26         function validatePassword(){
27             if(password.value != confirm_password.value) {
28                 confirm_password.setCustomValidity("Passwords Don't Match");
29             } else {
30                 confirm_password.setCustomValidity("");
31             }
32         }
33
34         password.onchange = validatePassword;
35         confirm_password.onkeyup = validatePassword;
```

Рисунок 3.4 – Код окна регистрации

3.3 Личный кабинет

Если жилец уже прошел регистрацию, ему достаточно авторизоваться и войти в личный кабинет.

Жильцы имеют возможность скачать в личном кабинете финансовый отчет по указанному ими при регистрации дому, в котором имеется информация о том, какие работы проводились в этом доме. Помимо этого, в личном кабинете представлена информация о пользователе.

На рисунках 3.5 и 3.6 представлена страница личного кабинета.

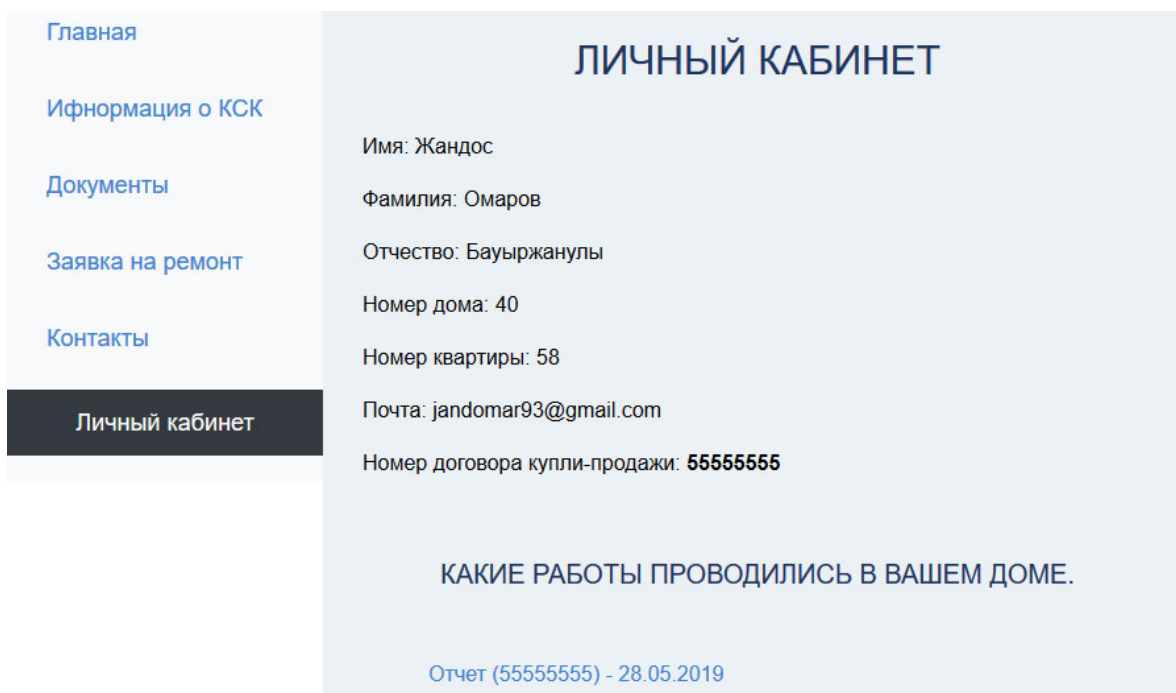


Рисунок 3.5 – Страница личного кабинета

```
1  {% extends 'article/base.html' %}
2  {% block title %}
3      Личный кабинет
4  {% endblock %}
5  {% block page_content %}
6      <h2>Личный кабинет</h2>
7
8      <p>Имя: {{ profile.first_name }}</p>
9      <p>Фамилия: {{ profile.last_name }}</p>
10     <p>Отчество: {{ profile.patronymic }}</p>
11     <p>Номер дома: {{ profile.home }}</p>
12     <p>Номер квартиры: {{ profile.apartment }}</p>
13     <p>Почта: {{ profile.email }}</p>
14     <p>Номер договора купли-продажи: <b>{{ profile.contract }}</b></p>
15
16     <h3>Какие работы проводились в Вашем доме.</h3>
17
18     <div class="docs">
19         {% for doc in docs %}
20             <p><a href="/media/{{ doc.file }}" target="_blank">{{ doc.name }} - {{ doc.date|date:"d.m.Y" }}</a></p>
21         {% endfor %}
22     </div>
23  {% endblock %}
```

Рисунок 3.6 – Код страницы личного кабинета

3.4 Восстановление пароля

В случае если пользователь забудет пароль существует страница восстановления пароля, с указанием почты, на которую приходит письмо.

На рисунках 3.7 и 3.8 представлена страница восстановления пароля.

The screenshot shows a web interface for password recovery. On the left is a navigation menu with links: Главная, Информация о КСК, Документы, Заявка на ремонт, Контакты, and a dark bar for Авторизация. Below the menu are input fields for *Email and *Пароль, and a Войти button. At the bottom of the menu are links for Регистрация and Забыли пароль?. The main content area is titled ВОССТАНОВЛЕНИЕ ПАРОЛЯ and contains a form with an input field labeled Введите ваш логин (E-mail) and a blue button labeled Восстановить пароль.

Рисунок 3.7 – Страница восстановления пароля

```
1  {% extends 'article/base.html' %}
2  {% block title %}
3      Восстановление пароля
4  {% endblock %}
5  {% block page_content %}
6      <h2>Восстановление пароля</h2>
7
8      <form method="POST" class="fb">
9          <input placeholder="Введите ваш логин (E-mail)">
10         <input type="submit" value="Восстановить пароль" class="login">
11     </form>
12  {% endblock %}
```


Рисунок 3.8 – Код страницы восстановления пароля

3.5 Заявка на ремонт

Жилец имеет возможность открыть страницу отправки заявок и заполнить открывшуюся форму, после чего на указанную почту автоматически придет письмо с сообщением об успешной отправке заявки.

На рисунках 3.9 и 3.10 представлена страница отправки заявок на ремонт.

Главная

Информация о КСК

Документы

Заявка на ремонт

Контакты

Авторизация

*Email

*Пароль

Войти

Регистрация

Забыли пароль?

ЗАЯВКА НА РЕМОНТ

Ваше имя

Номер квартиры

Телефон

Адрес электронной почты

Описание неисправности

Отправить заявку

Рисунок 3.9 – Форма ввода заявки на ремонт

```
1   {% extends 'article/base.html' %}
2   {% block title %}
3       Заявка на ремонт
4   {% endblock %}
5   {% block page_content %}
6       <h2>Заявка на ремонт</h2>
7
8       <form method="POST" class="fb">
9           <input placeholder="Ваше имя">
10          <input placeholder="Номер квартиры">
11          <input placeholder="Телефон">
12          <input placeholder="Адрес электронной почты">
13          <textarea rows="10" cols="45" placeholder="Описание неисправности"></textarea>
14          <input type="submit" value="Отправить заявку" class="login">
15      </form>
16   {% endblock %}
```

Рисунок 3.10 – Код страницы отправки заявки на ремонт

3.6 Документация

Для прозрачности ведения финансовой деятельности на сайт добавлена страница, хранящая различную документацию о КСК, например, информацию о смете расходов в виде файла word.

Страница документации представлена на рисунках 3.11 и 3.12.

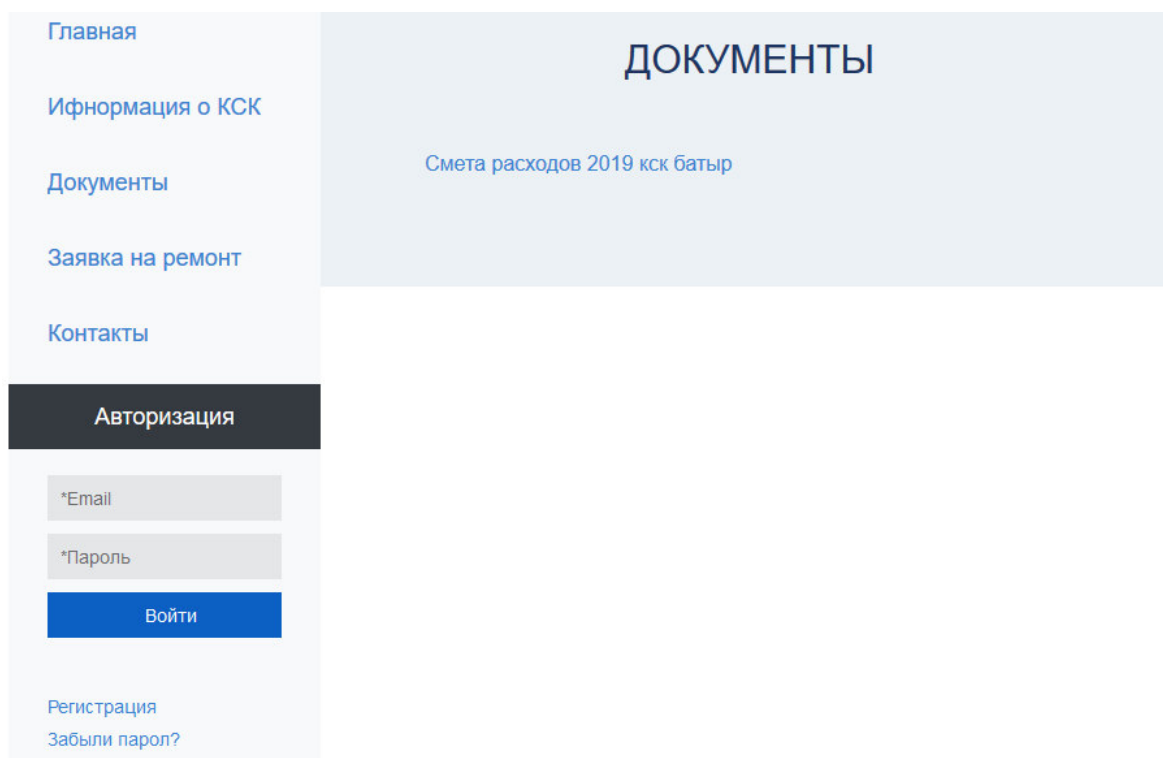


Рисунок 3.11 – Страница документов

```
1  {% extends 'article/base.html' %}
2  {% block title %}
3      Документы
4  {% endblock %}
5  {% block page_content %}
6      <h2>Документы</h2>
7      <div class="docs">
8          {% for doc in docs %}
9              <p><a href="/media/{{ doc.file }}" target="_blank">{{ doc.name }} - {{ doc.date|date:"d.m.Y" }}</a></p>
10             {% endfor %}
11     </div>
12  {% endblock %}
```

Рисунок 3.12 – Код страницы документов

3.7 Информация о кооперативе собственников квартир

На странице информации о КСК посетители сайта могут ознакомиться с деятельностью и управленческим составом кооператива. Перечислены адреса домов, обслуживаемые КСК «Батыр».

На рисунках 3.13 и 3.14 представлена страница информации о КСК.

The screenshot shows a web page with a navigation menu on the left and a main content area. The navigation menu includes links for 'Главная', 'Информация о КСК', 'Документы', 'Заявка на ремонт', 'Контакты', and 'Авторизация'. The main content area is titled 'ИНФОРМАЦИЯ О КСК:' and contains the following text:

Компания КСК «Батыр» находится по адресу, г. Алматы, Ауэзовский район, ул. Сулейменова д.17.

КСК «Батыр» обслуживает дома, находящиеся по адресу мкр-н Таугуль д.19, 40-41, 43-45, 47,49, 51-53, а также в мкр-не Таугуль-1 д.32-34, 41-52,85.

Основным предметом деятельности кооператива является: комплексное управление, эксплуатация и ремонт недвижимого имущества в многоквартирном доме, направленные на реализацию собственниками жилых и нежилых помещений своих прав по владению, пользованию и распоряжению общим имуществом домовладения. Для высокоэффективного управления жилым домом, выполнения задач по его качественной технической эксплуатации, ремонту, благоустройству, содержанию внутридомовой территории и паркинга.

Управление:

Общее собрание КСК «Батыр»;
Правление КСК «Батыр»;
Ревизионная комиссия КСК «Батыр».

At the bottom of the page, there are links for 'Регистрация' and 'Забыли пароль?'.

Рисунок 3.13 – Страница информации о КСК

```
1  {% extends 'article/base.html' %}
2  {% block title %}
3      Информация
4  {% endblock %}
5  {% block page_content %}
6      <h2>Информация о КСК</h2>
7      <p><strong>Уважаемые собственники!</strong></p>
8      <p>В настоящее время некоторыми собственниками нашего дома и их представителями распространяется очередная недостоверная информация относительно деятельности
9      <p>Как можно понять из длинного художественного повествования в рассылаемых письмах, претензии сводятся к нескольким вопросам:<br /><br />
10     отсутствие разрешительной документации на проведение ремонта кровли над 12 подъездом;<br />
11     незаконное использование ТСЖ общего имущества;<br />
12     парковка автотранспортных средств подрядных организаций во дворе дома.<br />
13     Со всей ответственностью заявляем, что все указанные претензии являются элементарным и безосновательным наветом.</p>
14     <p>По первому вопросу сообщаем, что по результатам обследования ГУП «МосжилНИИпроект» от 2010 года выдано заключение о том, что кровля нашего дома находится
15     <p>В 2014 году был разработан проект ремонта кровли, который согласован Департаментом культурного наследия г. Москвы 27.11.2014 № ДКН-16-08-103/4-30.</p>
16     <p>25.08.15 выдано разрешение Мосгорнаследия на производство работ по ремонту кровли.</p>
17     <p>30 октября 2015 года ОАТИ г. Москвы выдан ордер (разрешение) на производство работ с установкой лесов.</p>
18     <p>Для производства работ привлечена специализированная организация, имеющая лицензии на производство работ на объектах культурного наследия, выданную Минис
19  {% endblock %}
```

Рисунок 3.14 – Код страницы информации о КСК

3.8 Контакты

На странице контактов указаны наименование, адрес, телефон и режим работы КСК «Батыр».

На рисунках 3.15 и 3.16 представлена страница контактов.

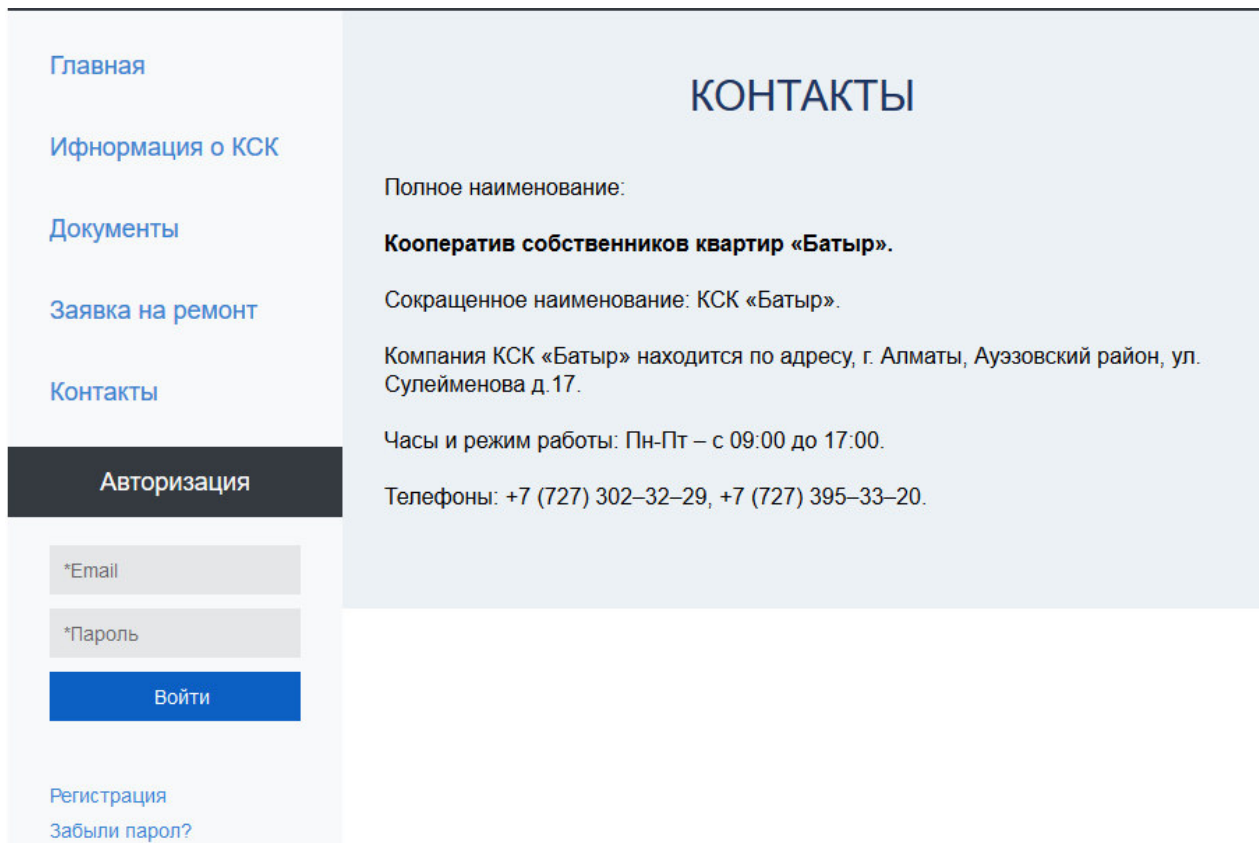


Рисунок 3.15 – Страница контактов

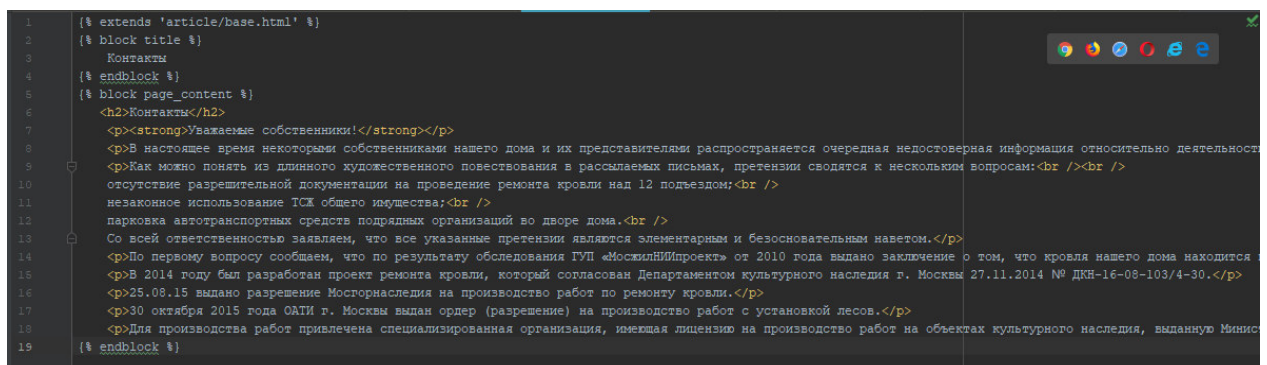


Рисунок 3.16 – Код страницы контактов

4 Экономическое обоснование разработки проекта

4.1 Трудоемкость разработки программного продукта

Тема дипломного проекта – «Разработка автоматизированной информационной системы службы КСК».

Перед экономическим обоснованием стоят следующие цели: рассчитать и проанализировать затраты, которые необходимы для создания и реализации автоматизированной ИС в виде веб-сайта, посчитать затраты на проектирование сайта и его функционирование; установить условия окупаемости затрат и их сроки; аргументировать рациональность проектирования сайта КСК.

В этой главе рассчитывается экономический расчет коммерческой цены на исследование проекта. Затраты на создание и внедрение информационной системы проекта учитываются в расчетах.

Для успешной реализации проекта необходимы трудовые, материальные и финансовые ресурсы.

Составлен план, длительностью в один месяц (четыре недели) с целью проектирования и реализации проекта, состоящий из трех этапов.

В первый этап входит:

- установка задачи;
- описание содержания работы.

Второй этап включает заключается в разработке программного продукта (ПП).

Третий этап включает:

- тестирование;
- отладку;
- внедрение проекта.

В таблице 4.1 представлены оценка их трудоемкости и разделение работ по этапам и видам.

Таблица 4.1 – Разделение работ по этапам и типам и оценка их трудоемкости

Этап разработки программного продукта	Тип работы в текущем этапе	Трудоемкость разработки ПП, чел.× ч.
Первый этап	Постановка задачи; Описание содержания работы	1×15
Второй этап	Разработка программного продукта	1×100
Третий этап	Тестирование; Отладка; Внедрение.	1×40
Итого трудоемкость выполнения дипломной работы		155

4.2 Расчет затрат на разработку программного продукта

Для того чтобы провести расчеты, необходимы как данные о затратах на ПП и оборудование, так и материальные затраты. Данные об этих затратах приведены в таблицах 4.2 и 4.3.

Таблица 4.2 – Расходы на ПО и оборудование

Название ресурса	Единица измерения	Количество использованного материала	Цена за единицу, тг	Сумма, тг
Современный офисный ПК	Шт	1	150000	150000
Струйный МФУ HP Deskjet 2630	Шт	1	20000	20000
Phyton	Шт	1	бесплатно	
Django	Шт	1	бесплатно	
SQLite	Шт	1	бесплатно	
Итоговые расходы на ПО и оборудование				170000

Таблица 4.3 – Материальные затраты

Название материального ресурса	Единица измерения	Количество использованного материала	Цена за единицу, тг	Сумма, тг
Пачка листов бумаги	Шт	2	1300	2600
Струйный картридж	Шт	1	1600	1600
Хостинг	мес.	6	2000	12000
Покупка домена	Год	1	3388	3388
Итоговые затраты на материальные ресурсы				19588

Общее число затрат на материальные ресурсы рассчитывается по формуле:

$$Z_m = \sum_{i=1}^n P_i \times C_i \quad (4.1)$$

где P_i - расход i -го материального ресурса, в натуральных единицах;
 C_i - цена за единицу i -го материального ресурса, в тенге;
 i - вид материального ресурса;

n - число видов материальных ресурсов.

$$Z_m = ((1300 \times 2) + 1600 + (6 \times 2000) + 3388) = 19588 \text{ тг.}$$

В таблице 4.4 приведена форма по которой надо посчитать затраты на электроэнергию.

Таблица 4.4 – Затраты на электроэнергию

Название Оборудования	Паспортная мощность, кВт	Коэффициент израсходованной мощности	Время работы оборудования для разработки ПП, ч	Цена электроэнергии, тг/кВт*ч	Сумма, тг
Современный офисный ПК	0,54	0,7	155	16,53	1125
Струйный МФУ HP Deskjet 2630	0,28	0,7	2	16,53	7
ИТОГО затраты на электроэнергию					1132

Окончательная сумма затрат на электроэнергию находится по формуле (4.2):

$$Z_э = \sum_{i=1}^n M_i * K_i * T_i * Ц \quad (4.2)$$

где i - вид электрооборудования;
n - количество электрооборудования;
M_i - паспортная мощность i-го электрооборудования, кВт;
K_i - коэффициент употребления мощности i-го электрооборудования (берется K_i=0.7,0.9);
T_i - время работы i-го оборудования за все время создания программного продукта, ч;
Ц - цена электроэнергии, тг/кВт*ч.

$$Z_{э1} = 0,54 \cdot 0,7 \cdot 180 \cdot 16,53 = 1125 \text{ тг.}$$

$$Z_{э2} = 0,28 \cdot 0,7 \cdot 2 \cdot 16,53 = 7 \text{ тг.}$$

$$Z_э = Z_{э1} + Z_{э2} = 1125 + 7 = 1132 \text{ тг.}$$

4.3 Трудовые ресурсы, задействованные в работе

Чтобы посчитать окончательную сумму затрат на оплату заработной платы используем формулу:

$$Z_{\text{тр}} = \sum_{i=1}^n ЧС_i \cdot T_i \quad (4.3)$$

где n – число разработчиков приложения;

i - категория работника;

ЧС_i - часовая ставка i-го работника, тг;

T_i – трудоемкость разработки программного продукта, чел.×ч.

Трудоемкость разработки ПП находится по данным таблицы 4.1.

Часовая ставка разработчика-программиста равна 1200 (тг/ч), а трудоемкость разработки – 155 часов (1 месяц).

В таблице 4.5 представлены итоги подсчета основной заработной платы.

Таблица 4.5 – Итоги подсчета затрат основной заработной платы

Название содержания работ	Исполнитель	Трудоемкость норма-час	Зарботная плата за час, тг/час	Сумма заработной платы, тг
Техническое задание	Разработчик	5	1200	12000
Моделирование ПП		10		36000
Программирование ПП		100		120000
Тестирование ПП		30		36000
Внедрение ПП		10		12000
Итого		155		216000

Далее следует расчет отчислений на социальный налог, составляющий 9,5% от дохода работника. Найдем социальные отчисления по следующей формуле:

$$Z_{\text{сзи}} = (Z_{\text{тр}} - Z_{\text{по}}) \cdot 0,095 \quad (4.4)$$

где Z_{по} – пенсионный отчисления, 10% от общего фонда оплаты труда, тенге. Рассчитаем пенсионные отчисления по следующей формуле:

$$Z_{\text{по}} = Z_{\text{тр}} \cdot 0,1 \quad (4.5)$$

Применяем формулы (4.4) и (4.5) и получаем:

$$Z_{\text{ПО}} = 216000 \cdot 0,1 = 21600 \text{ тг.}$$

$$Z_{\text{сзи}} = (216000 - 21600) \cdot 0,095 = 18468 \text{ тг.}$$

Сумма годовых амортизационных отчислений определяется по формуле:

$$A = \text{перв. стоимость} \cdot \text{норма амортизации}/100 \quad (4.6)$$

В таблице 4.6 приведены амортизационные отчисления.

Таблица 4.6 – Амортизация основных фондов

Название оборудования и ПО	Цена оборудования и ПО, тг	Годовая норма амортизации, %	Срок полезного применения оборудования и ПО, лет	Сумма амортизации в год, тг	Сумма амортизации в месяц, тг
Современный офисный ПК	150 000	20	5	30 000	2500
Струйный МФУ HP Deskjet 2630	20 000	20	5	4 000	333
Phyton	Распространяется бесплатно				
Django	Распространяется бесплатно				
SQLite	Распространяется бесплатно				
Итоговая амортизация основных фондов					2833

Годовые нормы амортизации основных фондов (ОФ) принимаются по налоговому кодексу Республики Казахстан или находятся, исходя из возможного срока полезного использования ОФ:

$$H_{Ai} = 100/T_{Ni} \quad (4.7)$$

где T_{Ni} - возможный срок использования i -го ОФ, год.

$$H_{\text{Аоб}} = \frac{100}{5} = 20$$

$$A_{\text{н}} = \frac{150000 \cdot 20}{100} = 30000 \text{ тг.}$$

$$A_{\text{м}} = \frac{20000 \cdot 20}{100} = 4000 \text{ тг.}$$

Сумма амортизации за один месяц = $A/12$.

Сумма амортизационных отчислений в месяц равна 2833 тенге.

В статью «Прочие затраты» входят затраты за арендную плату, в которую входят коммунальные платежи, канцелярские и прочие хозяйственные расходы.

Цена аренды помещения в месяц равна 50 000 тенге. В эту цену также входят коммунальные услуги.

По формуле (4.8) находится арендная плата:

$$AP = C_a \cdot S \quad (4.8)$$

где C_a – срок аренды;

S – стоимость аренды за один месяц.

$$AP = 50000 \cdot 1 = 50000 \text{ тг.}$$

Расходы на интернет, месячная стоимость которого равна 6000 тенге равны:

$$P_{и} = 6000 \cdot 1 = 6000 \text{ тг.}$$

Остальные хозяйственные расходы равны 10000 тенге.

Прочие затраты = 50000 + 6000 + 10000 = 66000 тг.

В таблице 4.7 приведена смета затрат на создание программного продукта.

Таблица 4.7 – Смета затрат на разработку программного продукта

Тип затрат	Сумма, тг
Материальные затраты:	
а) материалы	19588
б) электроэнергия	1132
Расходы на оплату труда	216000
Отчисления на социальные нужды	18468
Амортизация основных фондов	2833
Прочие расходы	66000
Итого	324021

4.4 Определение договорной цены программного продукта

Размер возможной (договорной) цены программного продукта должна устанавливаться с учетом эффективности, качества и сроков ее выполнения на уровне, отвечающем экономическим интересам заказчика (потребителя) и исполнителя.

Договорная цена (ДЦ) для прикладных программных продуктов рассчитывается по формуле:

$$Ц_{д} = З_{\text{нир}} \cdot \left(1 + \frac{P}{100}\right) \quad (4.9)$$

где $Z_{\text{нир}}$ – затраты на разработку ПП (из таблицы 4.7), тг;
 P – средний уровень рентабельности ПП, %.

$$Ц_{д} = 324021 \cdot \left(1 + \frac{20}{100}\right) = 388825,2 \text{ тг.}$$

Далее находим цену реализации с учетом налога на добавленную стоимость (НДС), ставка НДС устанавливается законодательно Налоговым Кодексом РК. На 2019 год ставка НДС равна 12%.

Цена реализации с учетом НДС находится по формуле:

$$\begin{aligned} Ц_{р} &= Ц_{д} + Ц_{д} \cdot \text{НДС} \\ Ц_{р} &= 388825,2 + 388825,2 \cdot 0,12 = 435485 \text{ тг.} \end{aligned} \quad (4.10)$$

4.5 Расчет срока окупаемости программного продукта

В результате использования веб-сайта КСК возможна экономия расходов на выплату зарплаты диспетчеру, т.к. серьезно уменьшится нагрузка на диспетчера. Благодаря нагрузке на диспетчера, возможно его увольнение и распределение его обязанностей на плечи председателя и бухгалтера КСК. Это увеличит заработную плату оставшихся сотрудников, но в сумме, с вычетом зарплаты диспетчера, сэкономит значительную сумму в долгосрочной перспективе.

Зарплата диспетчера составляет 80000 тг/мес. Надбавка к зарплате оставшихся двух сотрудников составляет 30000 тенге в сумме в месяц.

По формуле (4.11) находим расчетный срок окупаемости продукта:

$$T_{\text{ок}} = \frac{C}{\Pi} \quad (4.11)$$

где C - расходы на разработку и внедрение системы, тг; Π - прибыль, тг/месяц.

$$T_{\text{ок}} = \frac{435485}{80000 - 30000} = 8,7 \text{ месяцев}$$

В нашем случае срок окупаемости проекта равен 8,7 месяцев.

4.6 Оценка социально-экономических результатов

Социально-экономическим показателям функционирования ИС является обработка таких факторов как:

- фактор качества процесса управления;
- фактор длительности и срока проектирования ПП;
- фактор расходов на создание приложения и эксплуатацию ПП;
- количество разработчиков.

Эффективность ПП зависит от его качества и эффективности процесса создания и сопровождения. С точки зрения специалиста эффективность процесса создания определяется отсутствием нужды в мощном оборудовании. Компьютер средней мощности с доступом к интернету это все что необходимо специалисту для осуществления процесса создания. С точки зрения применения ресурсов пользователи не ограничены ничем, только необходимостью иметь доступ к интернету.

Разработка веб-сайта займет 155 часов (1 месяц). 15 часов из которых занимает процесс установки задачи и проектирование приложения, 100 - разработка, 40 - тестирование, отладка и внедрение. В процессе разработки участвует один человек – программист. Затраты на реализацию программного продукта составляют 324021 тенге. Договорная цена – 388825,2 тенге. Цена создания – 435485 тенге.

Пользование конечным программным продуктом не требует специальной подготовки или квалификации. С поставленной задачей справиться любой сотрудник, умеющий пользоваться веб-ресурсами.

5 Безопасность жизнедеятельности

5.1 Анализ условий труда

В дипломном проекте мы создаем автоматизированную информационную систему для службы КСК, которая находится в офисном помещении. В помещении имеется два рабочих места с установленными на них компьютерами и мониторами. Техническое оборудование установлено в помещении длиной $A = 8$ м, шириной $B = 6$ м и высотой потолка $H = 4$ м.

На рисунке 5.1 показана схема помещения.

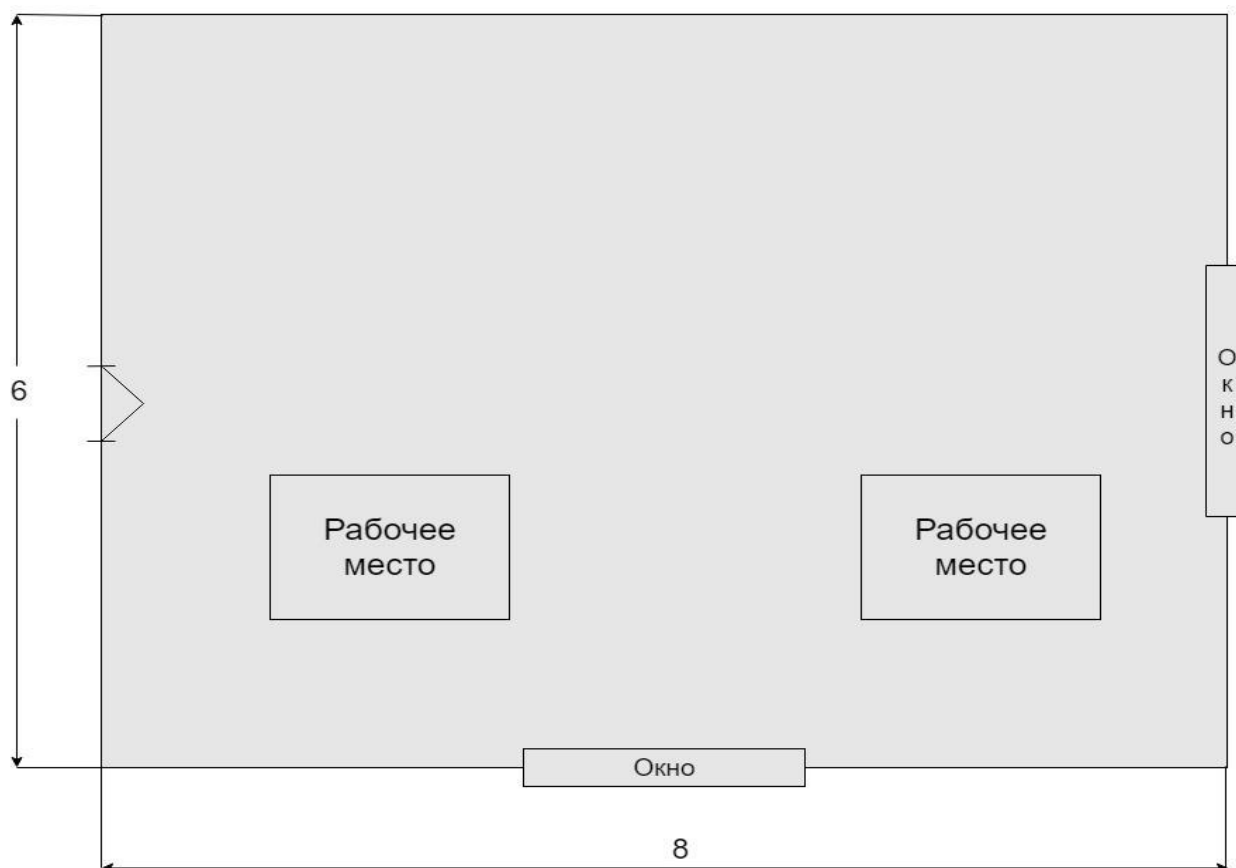


Рисунок 5.1 – Схема помещения

Помещение для работы должно соответствовать установленным нормам с целью организации благоприятных условий труда. На текущий момент помещение имеет размеры 8х6х4 м. Естественное освещение в помещении обеспечивают два окна, каждое окно имеет размеры равные 2,5 м². В помещении имеется не только естественный источник света, но и искусственные источники освещения, к которым относятся лампы накаливания мощностью 60 Вт. Приняты нормативы по $E_{\min} = 200$ лк, необходимые для расчета искусственного освещения. Уровень шума в комнате соответствует нормам, по причине отсутствия специальных производственных приборов и установок.

Расчет микроклимата выполняется в пространстве, которое находится на высоте до 2 метров над поверхностью уровня пола, на которой находится рабочее место работников. Устанавливаем категорию работ: в данном случае, определяется как легкая I,а. Благодаря содержанию водяных паров оценивается параметр влажности воздуха.

В текущее время условия труда не соответствуют нормам. Существует необходимость произвести реконструкцию аспирационной системы для создания благоприятных условий труда сотрудников. Для того чтобы все микроклиматические условия соответствовали требованиям «Санитарных норм, микроклимата производственных помещений», в целях нормализации работы оборудования необходима установка кондиционера.

В таблице 5.1 представлены показатели норм микроклимата.

Таблица 5.1 – Нормы микроклимата производственных помещений при выполнении работ средней тяжести

Категория работ	Период года	Температура, °С		Нормальная влажность, %		Скорость движения воздуха, м/с	
		Оптим.	Допус.	Оптим.	Допус.	Оптим.	Допус.
Лёгкая I,а	Холодный	18-20	17-23	40-60	75	0,2	0,1 не более
	Тёплый	21-23	18-27	40-60	65, при 26 °С	0,3	0,2-0,4

5.2 Расчет тепловых нагрузок в помещении

В помещениях различного назначения действуют в основном наружные тепловые нагрузки, возникающие снаружи помещения, а также внутренние тепловые нагрузки, возникающие внутри зданий.

$$Q_{огр} = V_{пом} \cdot X_0 \cdot (t_{Нрасч} - t_{Врасч}), \text{ Вт} \quad (5.1)$$

где $V_{пом}$ – объем помещения, м³.

X_0 – удельная тепловая характеристика, Вт/м³°С.

$$V_{пом} = 8 \cdot 6 \cdot 4 = 192 \text{ м}^3.$$

$$X_0 = 0,42 \text{ Вт/м}^3 \text{°С}.$$

$t_{Нрасч}$ – наружная температура (параметр А). Для холодного периода – средняя температура самого холодного месяца в 13 часов, для теплого периода – средней температуре самого жаркого месяца в 13 часов.

$t_{Врасч}$ – внутренняя температура, избирается с учетом комфортных условий или технологических требованиях, предъявляемых к производственным процессам.

Для теплого периода года:

$$t_{\text{Нрасч}} = 29,4 \text{ }^{\circ}\text{C}$$

$$t_{\text{Врасч}} = 24 \text{ }^{\circ}\text{C}$$

$$Q_{\text{огр}} = 192 \cdot 0,42 \cdot 5,4 = 435,46 \text{ Вт}$$

Для холодного периода года:

$$t_{\text{Нрасч}} = -9 \text{ }^{\circ}\text{C}$$

$$t_{\text{Врасч}} = 20 \text{ }^{\circ}\text{C}$$

$$Q_{\text{огр}} = 192 \cdot 0,42 \cdot | - 29 | = 2388,56 \text{ Вт}$$

В зависимости от типа стекла, избыточная теплота солнечного излучения почти до 90% поглощается средой помещения, а оставшаяся часть отражается. При максимальном уровне излучения, которое имеет прямую и рассеянную составляющие, достигается максимальная тепловая нагрузка. Такие показатели как ширина местности, времени года и времени суток влияют на интенсивность излучения.

По формуле (5.2) определяется теплоступление от солнечного излучения через остекление:

$$Q_p = (q^I \cdot F_0^I + q^{II} \cdot F_0^{II}) \cdot \beta_{\text{с.з.}} \quad (5.2)$$

где q^I , q^{II} – тепловые потоки от прямой и рассеянной солнечных радиаций, Вт/м²;

F_0^I , F_0^{II} – площади светового проема, облучаемые и необлучаемые прямой солнечной радиацией, м²;

$\beta_{\text{с.з.}}$ – коэффициент теплопропускания.

$$\beta_{\text{с.з.}} = 0,15$$

При отсутствии наружных затеняющих козырьков, ребер и т. д. для периода облучения остекления солнцем, когда его лучи проникают через окно в помещение $F_0^I = F_0$; $F_0^{II} = 0$:

$$Q_p = q^I \cdot F_0 \cdot \beta_{\text{с.з.}} = (q_{\text{вп}} + q_{\text{вр}}) \cdot K_1^C \cdot K_2 \cdot \beta_{\text{с.з.}} \cdot n \cdot S_0, \text{ Вт} \quad (5.3)$$

где F_0 – площадь светового проема (n – число окон; S_0 – площадь 1 окна);

$q_{\text{вп}}$; $q_{\text{вр}}$ – тепловые потоки от прямой рассеянной радиации, Вт/м²;

K_1 – коэффициент затемнения остекления переплетами (K_1^C – для облученных проемов);

K_2 – коэффициент загрязнения остекления.

$$q_{\text{вп}} = 63 \text{ Вт/м}^2;$$

$$q_{\text{вр}} = 65 \text{ Вт/м}^2;$$

$$F_o = nS_o = 2 \cdot 2,5 = 5 \text{ м}^2;$$

$$K_{C_1} = 0,72;$$

$$K_2 = 0,9.$$

Тогда:

$$Q_p = (63 + 65) \cdot 0,72 \cdot 0,9 \cdot 0,15 \cdot 5 = 62,2 \text{ Вт}$$

5.3 Расчет внутренних тепловых нагрузок в помещении

Внутренние нагрузки в офисных, жилых или относящихся к сфере обслуживания помещениях суммируются в основном из:

- тепла, выделяемого людьми;
- тепла, выделяемого лампами и электробытовыми, осветительными приборами;
- тепла, выделяемого печатающими устройствами, компьютерами, фотокопировальными машинами пр.

В технологических и производственных помещениях различного назначения дополнительными источниками тепловыделений могут стать: нагретое производственное оборудование, горячие материалы, в том числе жидкости и различного рода полуфабрикаты, продукты сгорания и химических реакций.

Согласно таблице 5.1 летом при 24⁰С один мужчина выделяет явного тепла 67 Вт, а общего – 102 Вт. Женщина выделяет 85% от нормы тепловыделений взрослого мужчины. Рассчитаем выделение явного тепла в помещении:

$$Q_{л}^я = 67 \cdot 2 \cdot 0,85 = 113,9 \text{ Вт}.$$

Посчитаем выделение общего тепла:

$$Q_{л}^o = 102 \cdot 2 \cdot 0,85 = 173,4 \text{ Вт}.$$

Согласно таблице 5.1 зимой при 20⁰С один мужчина выделяет явного тепла 82 Вт, а общего – 103 Вт. Рассчитаем выделение явного тепла в помещении составит:

$$Q_{з}^я = 82 \cdot 2 \cdot 0,85 = 139,4 \text{ Вт}.$$

Посчитаем выделение общего тепла:

$$Q_{з}^o = 103 \cdot 3 + 103 \cdot 3 \cdot 0,85 = 571,65 \text{ Вт}.$$

Теплопоступление от ламп высчитывается по формуле:

$$Q_{\text{осв}} = \eta \cdot N_{\text{осв}} \cdot F_{\text{пол}}, \text{ Вт} \quad (5.4)$$

где η – коэффициент перехода электрической энергии в тепловую (для люминесцентных ламп $\eta=0.5-0.6$);

$N_{\text{осв}}$ – установленная мощность ламп ($N=60 \text{ Вт/м}^2$);

$F_{\text{пол}}$ – площадь пола.

$$F_{\text{пол}} = 8 \cdot 6 = 48 \text{ м}^2$$

$$Q_{\text{осв}} = 0,5 \cdot 60 \cdot 48 = 1440 \text{ Вт}.$$

По формуле (5.5) рассчитывается тепло, выделяемое производственным оборудованием:

$$Q_{\text{об}} = N_{\text{уст}} \cdot K \quad (5.5)$$

$$Q_{\text{об}} = 1,8 \cdot 2 \cdot 0,95 = 3,42 \text{ кВт}$$

Теплопритоки, появляющиеся из-за имеющейся оргтехники, составляют 30% мощности оборудования:

$$Q_{\text{орг}} = 1,8 \cdot 2 \cdot 0,3 = 1,08 \text{ кВт}$$

5.4 Расчет теплового баланса помещения

По результатам осуществленных расчетов составим баланс теплоступлений в помещении:

Летом:

$$Q_{\text{изб}} = 62,2 + 113,9 + 1440 + 3420 + 1080 + 435,46 = 6551,56 \text{ Дж}$$

Зимой:

$$Q_{\text{изб}} = 62,2 + 139,4 + 1440 + 3420 + 1090 + 2388,56 = 8540,16 \text{ Дж}$$

Учитывая, что тепловой баланс зимой больше летнего теплового баланса, то посчитаем теплонапряженность воздуха по формуле:

$$Q_{\text{н}} = \frac{Q_{\text{изблето}} \cdot 860}{V_{\text{пом}}}, \text{ ккал/м}^3 \quad (5.6)$$

$$Q_{\text{н}} = \frac{6551,56 \cdot 860}{192} = 29,35 \text{ ккал/м}^3$$

При $Q_{\text{н}} > 20 \text{ ккал/м}^3$, $\Delta t = 8 \text{ }^\circ\text{C}$,

Нахождение количества воздуха, нужного для поступления в помещение, рассчитывается по формуле:

$$L = \frac{Q_{изб} \cdot 860}{C \cdot \Delta t \cdot \gamma}, \text{ м}^3/\text{час} \quad (5.7)$$

$$L = \frac{6551,56 \cdot 860}{0,24 \cdot 8 \cdot 1,206 \cdot 10^4} = 243,33 \text{ м}^3/\text{час}$$

где $C=0,24$ ккал/(кг $^{\circ}$ С) – теплоемкость воздуха,
 $\gamma=1,206 \cdot 10^4$ кг/м 3 – удельная масса приточного воздуха.

Определение кратности воздухообмена:

$$N = \frac{L}{V_{пом}}, \text{ час}^{-1} \quad (5.8)$$

$$N = \frac{243,33}{192} = 1,28 \text{ час}^{-1}$$

5.5 Выбор кондиционера. Схема расположения

Опираясь на результаты вычислений, выбор падает на прецизионный кондиционер с верхней подачей воздуха модели SUA 0151, удовлетворяющий найденным требованиям по нужному количеству воздуха.



Рисунок 5.2 – Схема расположения кондиционера в производственном помещении

5.6 Вывод раздела

В результате анализа полученных данных делается вывод, что настоящая вентиляционная система не соответствует полностью требуемым нормам, в связи с чем, были посчитаны наружные и внутренние тепловые нагрузки в помещении. По результатам выбор пал на модель кондиционера с соответствующими характеристиками. По результатам расчетов ясно, что при достаточно большом пространстве для малого количества человек и оборудования, количество избыточного тепла считается малым, в следствии чего принято решение установить слабую систему кондиционирования.

От систем аспирации, вентиляции, отопления и кондиционирования воздуха зависит обеспечение воздушного комфорта в жилых и производственных помещениях. Задача кондиционирования воздуха заключается в вентиляции и отоплении, в поддержки таких условий воздушной среды, при которых каждый человек благодаря своей индивидуальной системе автоматической терморегуляции организма мог почувствовать себя комфортно, не замечая влияния этой среды.

Заключение

Дипломный проект посвящен разработке автоматизированной информационной системе службы КСК. В качестве информационной системы выбран веб-сайт.

В рамках дипломного проекта были изучены материалы по разработке сайта и его основные этапы разработки.

На основании вышеперечисленных исследований были определены цели и задачи, а также функции разрабатываемой автоматизированной информационной системы. Были сформулированы требования, предъявляемые к информационной системе.

Сделан выбор наиболее подходящих языка программирования, фреймворка, а также базы данных, при помощи которых был разработан веб-сайт КСК.

Сайт службы КСК состоит из следующих страниц: страница «Главная страница», страница «Регистрация», страница «Личный кабинет», страница «Восстановление пароля», страница «Контакты», страница «Информация о КСК», страница «Заявка на ремонт», страница «Документы». В каждом из разделов сайта содержится соответствующая названию страницы информация. Доступны для скачивания сметы и отчеты, составленные работниками КСК «Батыр».

Сайт позволит оставить заявку в службу КСК в любое время дня и день недели. Жильцы смогут посмотреть документацию по финансовым делам и затратам КСК, получают информацию о проведенных работах по их дому, указанному при регистрации. Информация о проведенных работах в по дому будут доступны только их жильцам благодаря автоматизированной регистрации с помощью номера договора купли-продажи квартиры, известный только КСК и хозяевам квартир.

В рамках выполнения дипломного проекта были произведены расчеты по выявлению затрат на проектирование и разработку веб-сайта, а также его рентабельности и эффективности.

Также было произведено описание рабочего места, рассмотрены вопросы безопасности жизнедеятельности, представлены нормы искусственного и естественного освещения, рассмотрен уровень шума. Найдена норма кондиционирования для рабочего помещения. Проведены расчеты тепловых нагрузки и баланса.

По итогам проекта видится улучшение взаимосвязи между жильцами и работниками КСК. Появился удобный способ оставить заявку на ремонт, в любое время дня и день недели. Большим шагом является прозрачность доходов и затрат производимые работниками КСК.

Список литературы

- 1 Балдин К.В., Уткин В.Б. «Информационные системы в экономике», издательство «Дашков и К», 2012 г. – 395 с.
- 2 Голицына О.Л., Партыка Т.Л., Попов И.И. «Системы управления базами данных. Учебное пособие», издательство «Инфра-М», 2011 г. – 431 с.
- 3 Золотов С.Ю. «Проектирование информационных систем», издательство «Эль Контент», 2013 г. – 86 с.
- 4 Карпова И.П. «Базы данных. Учебное пособие», издательство «Питер», 2013 г. – 240 с.
- 5 Гарсиа-Молина Гектор, Уидом Дженнифер, Ульман Джеффри Д. «Системы баз данных. Полный курс», издательство «Вильямс», 2017 г.
- 6 Новиков Б.А., Горшкова Е.А. «Основы технологии баз данных», издательство «ДМК Пресс», 2019 г.
- 7 Шелухин О.И. «Моделирование информационных систем. Учебное пособие», издательство «Горячая линия – Телеком», 2012 г.
- 8 Черемных С.В., Семенов И.О., Ручкин В.С. «Моделирование и анализ систем. IDEF-технологии», издательство «Финансы и статистика», 2006 г.
- 9 Федотова Е.Л. «Информационные технологии и системы. Учебное пособие», издательство «ИД форум», 2013 г.
- 10 Великович Л.С., Цветкова М.С. «Информатика и ИКТ», издательство «Инфра-М», 2012г.
- 11 Хлебников А.А. «Информационные технологии. Учебник», издательство «КноРус», 2014 г.
- 12 Гэддис Тони «Начинаем программировать на Python», издательство «БХВ-Петербург», 2019 г.
- 13 Бэрри Пол «Изучаем программирование на Python», издательство «Эксмо», 2017 г.
- 14 Васильев А.Н. «Python на примерах. Практический курс по программированию», издательство «НиТ», 2019 г.
- 15 Лутц Марк «Изучаем Python», 4-е издание, издательство «Символ-Плюс», 2011 г.
- 16 Седер Наоми «Python. Экспресс-курс», издательство «Питер», 2019 г.
- 17 Златопольский Д. «Основы программирования на языке Python», издательство «ДМК Пресс», 2018 г.
- 18 Любанович Билл «Простой Python. Современный стиль программирования», издательство «Питер», 2019 г.
- 19 Себеста Р. «Основные концепции языков программирования», издательство «Эксмо», 2011 г.
- 20 Торгашев А.Б. «Базы данных. Курс лекций для студентов, обучающихся по специальности информационные системы в экономике», 2011 г.
- 21 Калюжный Е.А., Михайлова С.В., Напреев С.Г. «Безопасность

жизнедеятельности. Учебное пособие», издательство «АГПИ», 2012 г.

22 Оппель Эндрю Дж., Вайнберг Пол Н., Грофф Джеймс Р. «SQL. Полное руководство», издательство «Вильямс», 2018 г.

23 Бейли Линн «Изучаем SQL», издательство «Питер», 2012 г.

24 Грекул В. «Проектирование информационных систем»
[Электронный ресурс]: <http://www.intuit.ru/studies/courses/2195/55/info>

25 Python [Электронный ресурс]: <http://www.python.org>

26 Django: The web framework [Электронный ресурс]:
<https://www.djangoproject.com>

27 SQLite [Электронный ресурс]: <https://www.sqlite.org/index.html>