

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра информационных и управляющих систем
Направление подготовки 09.03.01 – Информатика и вычислительная техника

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой

_____ А.В. Бушманов

« _____ » _____ 2024 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка финансового веб-приложения для стоматологической
клиники ООО «Зубная архитектура»

Исполнитель _____ Д.С. Беднова
студент группы 0103-об (подпись, дата)

Руководитель _____ А.В. Бушманов
доцент, канд. техн. наук (подпись, дата)

Консультант: _____ А.Б. Булгаков
по безопасности и экологичности (подпись, дата)
доцент, канд. техн. наук

Нормоконтроль _____ В.Н. Адаменко
инженер кафедры (подпись, дата)

Благовещенск 2024

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук

Кафедра информационных и управляющих систем

УТВЕРЖДАЮ

Зав. кафедрой

_____ А.В. Бушманов

«_____» _____ 2023 г.

ЗАДАНИЕ

К выпускной квалификационной работе студента Беднова Д.С.

1. Тема выпускной квалификационной работы: Разработка финансового веб-приложения для стоматологической клиники ООО «Зубная архитектура»
(утверждено приказом от 03.04.2024 № 890-уч)

2. Срок сдачи студентом законченной работы: _____

3. Исходные данные к бакалаврской работе: отчет о прохождении преддипломной практики, специальная литература, нормативные документы.

4. Содержание выпускной квалификационной работы: анализ деятельности предприятия; проектирование автоматизированной системы; разработка веб-приложения; инфологическое проектирование; логическое проектирование; физическое проектирование; создание базы данных для автоматизированной системы; тестирование готового веб-приложения.

5. Перечень материалов приложения: техническое задание.

6. Консультанты по выпускной квалификационной работе: Консультант по безопасности и экологичности: доцент, канд. техн. наук А.Б. Булгаков

7. Дата выдачи задания: 02.10.2023 года.

Руководитель выпускной квалификационной работы: доцент А.В. Бушманов

Задание принял к исполнению: 02.10.2023 г.: _____

(подпись студента)

РЕФЕРАТ

Бакалаврская работа содержит 74 с., 36 рисунков, 8 таблицы, 1 приложение, 11 источников.

АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ, PYTHON, ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ, РАЗРАБОТКА БАЗЫ ДАННЫХ, ЛОГИЧЕСКАЯ МОДЕЛЬ БД, ФИЗИЧЕСКАЯ МОДЕЛЬ БД, РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ

Цель работы: Разработка финансового веб-приложения для стоматологической клиники ООО «Зубная архитектура»

Выполнение проекта включает три этапа:

Первым этапом является исследование предметной области.

На втором этапе выполняется проектирование базы данных, состоящее из проектирования инфологической модели базы данных, логической модели и физического проектирования базы данных на выбранной СУБД.

В качестве СУБД для реализации базы данных была использована встроенная функция создания на Django.

Третьим этапом является разработка веб-приложения.

Результатом выполнения работы является веб-приложение на языке Python, связанное с базой данных, которое отражает деятельность стоматологической клиники ООО «Зубная архитектура»

СОДЕРЖАНИЕ

Введение	7
1 Анализ предметной области проектирования	9
1.1 Общие сведения о предприятии	9
1.2 Цели и задачи стоматологической клиники «Зубная Архитектура»	10
1.3 Организационная структура стоматологической клиники «Зубная Архитектура»	11
1.4 Документооборот стоматологической клиники «Зубная Архитектура»	13
1.5 Нормативные документы	15
1.6 Анализ существующих аналогов	16
1.7 Требования к функциям приложения	22
2 Проектирование информационной системы	24
2.1 Информационное проектирование базы данных	24
2.1.1 Сущности и их краткая характеристика	24
2.1.2 Описание связей между сущностями	26
2.1.3 Нормализация отношений	28
2.2 Логическое и физическое проектирование базы данных	28
3 Разработка программного продукта	31
3.1 Назначение и цели создания	31
3.2 Разработка технического задания	31
3.3 Функциональные требования	32
3.4 Техническое обеспечение	33
3.5 Работа с Django на Python	36
3.5.1 Написание моделей и представлений	36
3.5.2 Написание шаблонов	39
3.5.3 Кэширование и отслеживания ошибок	43

3.6 Работа с запросами и рассылками	45
3.6.1 Создание записи на прием	45
3.6.2 Создание рассылки по расписанию	47
4 Описание и тестирование программного продукта	49
4.1 Врачи и распределение по кабинетам	49
4.2 Пациенты каждого врача	49
4.3 Создание, редактирование и удаление записей	50
4.4 Поиск записей по критериям	51
5 Безопасность и экологичность	52
5.1 Безопасность	52
5.2 Экологичность	57
5.3 Чрезвычайные ситуации	58
Заключение	61
Библиографический список	63
Приложение А	65

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

АС – Автоматизированная система

БД – База данных

ПО – Программное обеспечение

АСОИУ – Автоматизированная система обработки информации и управления

СУБД – Система управления базами данных

ТЗ – Техническое задание

URL – Адрес веб-страницы

HTML – Язык разметки гипертекста

ВВЕДЕНИЕ

В современном мире информационные технологии (ИТ) играют критическую роль в повседневной деятельности различных сфер, и медицина не является исключением. В стоматологии, как и в любой другой медицинской области, внедрение автоматизированных систем имеет огромное значение для оптимизации процессов и улучшения обслуживания пациентов. В контексте производственной практики в ООО "Зубная архитектура" осуществлялась разработка финансового веб-приложения, целью которого являлось упрощение финансового учета и управления финансовыми потоками, а также обеспечение удобного доступа к информации.

Это веб-приложение разработано с учетом специфических потребностей стоматологической клиники и предлагает широкий спектр функциональных возможностей. Одной из главных задач приложения является автоматизация процессов финансовой отчетности, что позволяет значительно сократить время, затрачиваемое на подготовку отчетов, и снизить риск возникновения ошибок. Кроме того, оно обеспечивает контроль за финансовыми потоками в режиме реального времени, что позволяет оперативно реагировать на изменения и принимать обоснованные решения.

Одним из ключевых преимуществ данного приложения является его удобный и прозрачный интерфейс, который облегчает работу персонала клиники и повышает эффективность управления финансами. Благодаря интуитивно понятному интерфейсу пользователи могут быстро получать необходимую информацию и анализировать данные без необходимости в дополнительном обучении.

В процессе разработки были успешно реализованы следующие ключевые функции:

- возможность авторизации и регистрации пользователей: пользователи имеют возможность безопасно авторизоваться в системе и зарегистрироваться, что обеспечивает защиту доступа к данным;
- добавление, редактирование и удаление записей: приложение позволяет оперативно вносить изменения в базу данных клиники, что существенно упрощает управление информацией;
- поиск по различным критериям: пользователи могут осуществлять поиск по различным параметрам, таким как дата, имя врача, имя пациента, что облегчает навигацию и поиск необходимой информации;
- добавление возможности клиентам оставлять заявки на запись: это облегчить запись для клиентов, а также упростить работу с клиентами для персонала;
- рассылка отчетов о занесенных записях на указанные почтовые адреса: приложение позволяет автоматически рассылать отчеты о занесенных записях на указанные почтовые адреса, обеспечивая внутреннюю отчетность в клинике;
- кеширование данных: для оптимизации производительности приложения и снижения нагрузки на сервер используется кеширование данных;
- система отслеживания ошибок: реализована система отслеживания ошибок, которая позволяет оперативно выявлять и устранять проблемы в работе приложения.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ ПРОЕКТИРОВАНИЯ

1.1 Общие сведения о предприятии

Современный инновационный стоматологический центр "Зубная архитектура" известен своим широким спектром качественных стоматологических услуг, предоставляемых как жителям, так и гостям города.

В ее арсенале — все основные направления стоматологии: терапевтическое лечение, отбеливание зубов, чистка, восстановление и удаление зубов, имплантация, протезирование, а также коррекция прикуса.

Клиника имеет несколько филиалов, что позволяет ей расширить спектр предоставляемых услуг. Благодаря этому пациенты могут получать различные медицинские услуги в разных районах города, не теряя при этом в качестве обслуживания.

Клиника оснащена современным оборудованием и использует передовые технологии для обеспечения высокого качества медицинских услуг.

Специалисты "Зубной Архитектуры" регулярно проходят курсы повышения квалификации в России и за рубежом, что позволяет им быть в курсе последних достижений в области стоматологии.

Кроме того, клиника имеет собственную зуботехническую лабораторию, что позволяет ускорить процесс изготовления и установки зубных протезов и других стоматологических конструкций.

Юридический адрес:

675000, Амурская область, г. Благовещенск, Игнатьевское шоссе, 15 (2-й мкр.);

ИНН: 2801145322;

Телефоны: +7(4162)37-15-15, 21-15-15;

Факс: +7(4162)37-61-54;

E-mail: bykov0808@yandex.ru

1.2 Цели и задачи стоматологической клиники «Зубная Архитектура»

Объектом исследования является стоматологическая клиника «Зубная Архитектура», которая занимается рядом важных задач, связанных с улучшением стоматологического здоровья и предоставлением высококачественных медицинских услуг.

Цели стоматологической клиники «Зубная Архитектура»:

- обеспечение пациентам доступа к современным методам лечения и профилактики зубных заболеваний;
- пропаганда и реализация программ профилактики и ранней диагностики стоматологических заболеваний;
- проведение регулярных обучающих семинаров и тренингов для врачей и медицинского персонала;
- создание уютной и дружелюбной атмосферы в клинике, ориентированной на максимальное удобство и удовлетворенность клиентов;
- использование передового оборудования и современных методик лечения для повышения эффективности и безопасности процедур.

Задачи стоматологической клиники «Зубная Архитектура»:

- проведение комплексной диагностики, включая рентгенологические исследования, для постановки точного диагноза и назначения эффективного лечения;
- организация регулярных профилактических осмотров и профессиональной чистки зубов;
- коррекция прикуса и выравнивание зубов с использованием брекетов и капш;
- проведение процедур по отбеливанию зубов, реставрации и установке виниров для улучшения внешнего вида улыбки;
- выполнение сложных хирургических вмешательств, включая удаление зубов мудрости, имплантацию и реконструктивные операции;

- консультирование по вопросам гигиены полости рта и правильного ухода за зубами;
- ведение медицинской документации, управление записью пациентов и обеспечение их информационной поддержки;
- разработка и реализация маркетинговых стратегий для привлечения новых пациентов и повышения лояльности существующих.

Эти цели и задачи направлены на обеспечение высокого уровня медицинского обслуживания, удовлетворение потребностей пациентов и устойчивое развитие клиники.

1.3 Организационная структура стоматологической клиники «Зубная Архитектура»

Для того чтобы в полной мере осветить деятельность клиники и организацию ее работы, необходимо рассмотреть ее организационную структуру, представленную на рисунке 1.

В этой структуре выделяются ключевые должностные лица, обеспечивающие эффективное функционирование всего предприятия.

Каждые в организационной структуре имеют свои обязанности и цели, но вместе они образуют слаженную систему, направленную на обеспечение высокого уровня сервиса и удовлетворения потребностей пациентов.

Руководители и сотрудники каждого отдела работают в тесном взаимодействии, чтобы обеспечить эффективное функционирование клиники и достижение поставленных целей.

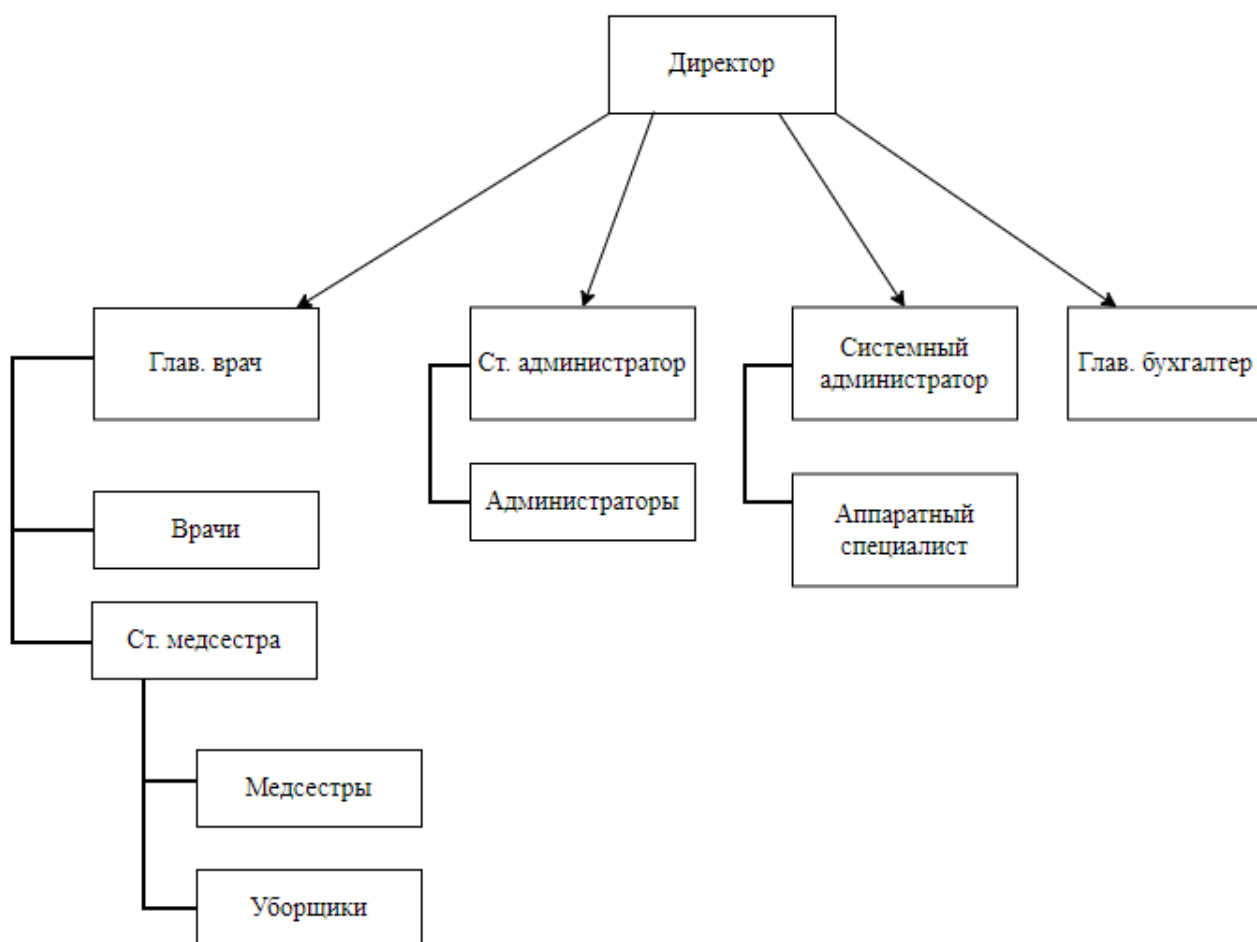


Рисунок 1 – Организационная структура предприятия

Клиника специализируется на оказании медицинских услуг в области стоматологии, где главными участниками процесса являются врачи и их ассистенты. Врачи работают в различных направлениях лечения зубов и их вариациях, в зависимости от специализации и опыта. Им помогают медсестры (ассистенты), которые оказывают помощь во время лечения, обеспечивают необходимые инструменты и материалы, а также следят за комфортом и безопасностью пациентов во время приема.

Главный бухгалтер выполняет ключевую роль в финансовой деятельности клиники. Он осуществляет учет всех финансовых операций, следит за движением средств, составляет отчеты о финансовом состоянии клиники и предоставляет необходимую информацию руководству. Кроме того, главный бухгалтер следит за соответствием финансовой деятельности клиники уста-

новленными нормам и законодательству.

Администраторы клиники отвечают за организацию работы приемной. Они занимаются записью пациентов на прием, их оформлением для получения медицинских услуг, а также предоставляют информацию о работе клиники и услугах, оказываемых в ней.

Системный администратор и аппаратный специалист, хотя и не являются постоянными сотрудниками клиники, играют важную роль в обеспечении функционирования информационных систем и технического оборудования. Системный администратор отвечает за настройку программного обеспечения, работу с компьютерами и обеспечение безопасности информации. Аппаратный специалист занимается установкой нового оборудования, его обслуживанием и ремонтом.

Руководство клиники, в лице директора "Зубной Архитектуры", обладает полномочиями по принятию стратегических решений и организации работы клиники в целом. Директор решает вопросы, связанные с финансовой, организационной и кадровой деятельностью, а также осуществляет контроль за выполнением поставленных задач.

1.4 Документооборот стоматологической клиники «Зубная Архитектура»

Для понимания информационных процессов на предприятии необходимо создать и описать систему внешнего и внутреннего документооборота. Внешний документооборот включает движение документов в правовом поле, где различные субъекты права – физические и юридические лица, граждане, предприятия и организации, органы местного самоуправления и государственной власти – взаимодействуют и осуществляют правовые отношения.

Это взаимодействие может происходить как между однотипными субъектами, так и с представителями других категорий. Диаграмма внешнего документооборота показана на рисунке 2.

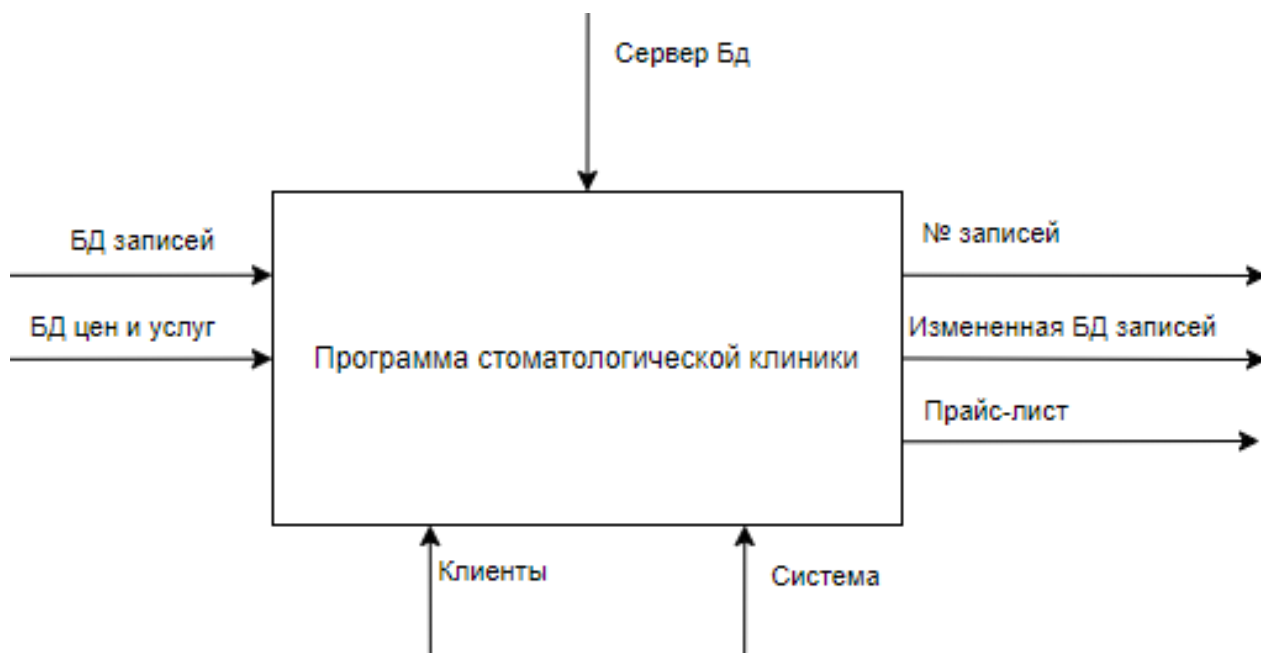


Рисунок 2 – Внешний документооборот

Внутренний документооборот представляет собой процесс управления и перемещения документов внутри компании или организации.

Этот процесс включает в себя создание, регистрацию, обработку, передачу, хранение и уничтожение документов, необходимых для функционирования предприятия.

Документооборот осуществляется в соответствии с внутренними нормативными актами, корпоративными правилами и процедурами, которые обеспечивают правильное и своевременное выполнение всех этапов работы с документами.

Диаграмма внутреннего документооборота, представленная на рисунке 3, визуализирует последовательность и взаимосвязь всех этапов обработки документов.

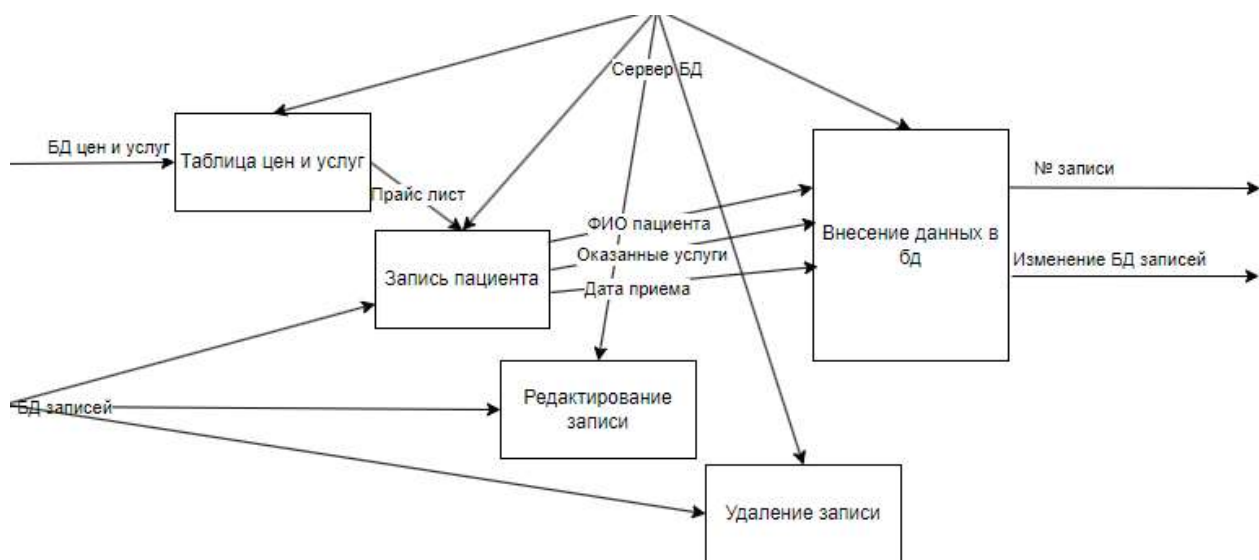


Рисунок 3 – Внутренний документооборот

1.5 Нормативные документы

Нормативные документы для стоматологической клиники «Зубная Архитектура» включают в себя такие типы, как:

- лицензии и сертификаты: например, лицензия на медицинскую деятельность, сертификаты качества, сертификаты соответствия стандартам безопасности;
- нормативные акты: это могут быть законы, постановления, постановления правительства, регламентирующие правила и стандарты оказания стоматологических услуг;
- протоколы и инструкции: протоколы обследования, лечения, профилактики, стерилизации инструментов и оборудования, а также инструкции по их применению;
- медицинские стандарты и рекомендации: например, стандарты антисептики и асептики в стоматологии, рекомендации по профилактике и лечению заболеваний полости рта;
- положения и регламенты: внутренние документы, регулирующие организационные процессы в клинике, включая положения о структуре и

функциях отделов, права и обязанности персонала и пациентов, регламенты приема и обслуживания пациентов;

- договоры и соглашения: например, договоры с пациентами, с поставщиками оборудования и расходных материалов, соглашения о конфиденциальности и защите персональных данных;

- финансовые документы: бухгалтерская отчетность, бюджеты, сметы расходов и доходов, финансовые планы и отчеты;

- документация по обучению и сертификации персонала: дипломы, сертификаты специализации, программы и отчеты обучения, аттестационные листы;

- документы по безопасности и экологии: планы мероприятий по обеспечению пожарной безопасности, санитарно-гигиенические нормы, экологические сертификаты и т. д;

- медицинская документация: истории болезни пациентов, протоколы осмотров, результаты анализов и обследований, документация по проведенным манипуляциям и операциям.

Эти указанные документы созданы для обеспечения соблюдения законности, повышения эффективности и согласования работы стоматологической клиники, а также для обеспечения соответствия её действий основным целям и задачам.

1.6 Анализ существующих аналогов

При анализе аналогов основными критериями, на которые стоит обратить внимание, являются следующие:

- гибкость и настраиваемость: при выборе аналога необходимо учитывать его способность к гибкой настройке и возможность доработки под уникальные потребности организации. Важно, чтобы программа позволяла адаптировать свои функции согласно конкретным требованиям бизнес-процессов;

– удобство использования: эффективное использование программы обеспечивается ее интуитивно понятным интерфейсом и удобством в работе. Пользователи должны легко освоить функционал и оперативно приступить к выполнению задач;

– возможность совместной работы: в современном бизнес-окружении важно иметь возможность работать над данными совместно с другими пользователями. Поэтому стоит выбирать программу, которая обеспечивает поддержку совместной работы и удобный обмен данными между участниками проекта;

– финансовые аспекты: при выборе аналога Excel важно учитывать бюджетные ограничения. Некоторые аналоги предлагаются бесплатно, в то время как другие могут требовать платную лицензию. Необходимо подобрать решение, которое соответствует финансовым возможностям организации и при этом обеспечивает необходимый функционал и качество работы.

В качестве подходящих аналогов были собраны такие программы, как:

В качестве аналогов программного средства выбраны:

- веб-приложение Google Sheets;
- программное средство LibreOffice Calc;
- программное средство OpenOffice Calc.

Рассмотрим все аналоги поочередно:

Google Sheets - это веб-приложение, разработанное Google для создания и редактирования электронных таблиц. Оно доступно через сервис Google Drive, что позволяет пользователям легко сохранять и синхронизировать свои таблицы с облачным хранилищем. Одним из ключевых преимуществ Google Sheets является возможность одновременной работы нескольких пользователей над таблицей в режиме реального времени, что способствует эффективной совместной работе.

Это приложение предоставляет обширный набор инструментов для форматирования данных, включая возможности изменения шрифтов, цветов

и стилей ячеек.

Кроме того, Google Sheets обладает встроенными функциями для выполнения различных расчетов, что делает его удобным инструментом для работы с числовыми данными. Возможность создания графиков и диаграмм позволяет визуализировать данные, делая их более понятными и наглядными.

Google Sheets также интегрируется с другими сервисами Google, что обеспечивает дополнительные возможности для управления данными и их обработки. Например, данные из таблиц могут быть легко импортированы в другие сервисы, такие как Google Docs или Google Slides, для создания отчетов или презентаций.

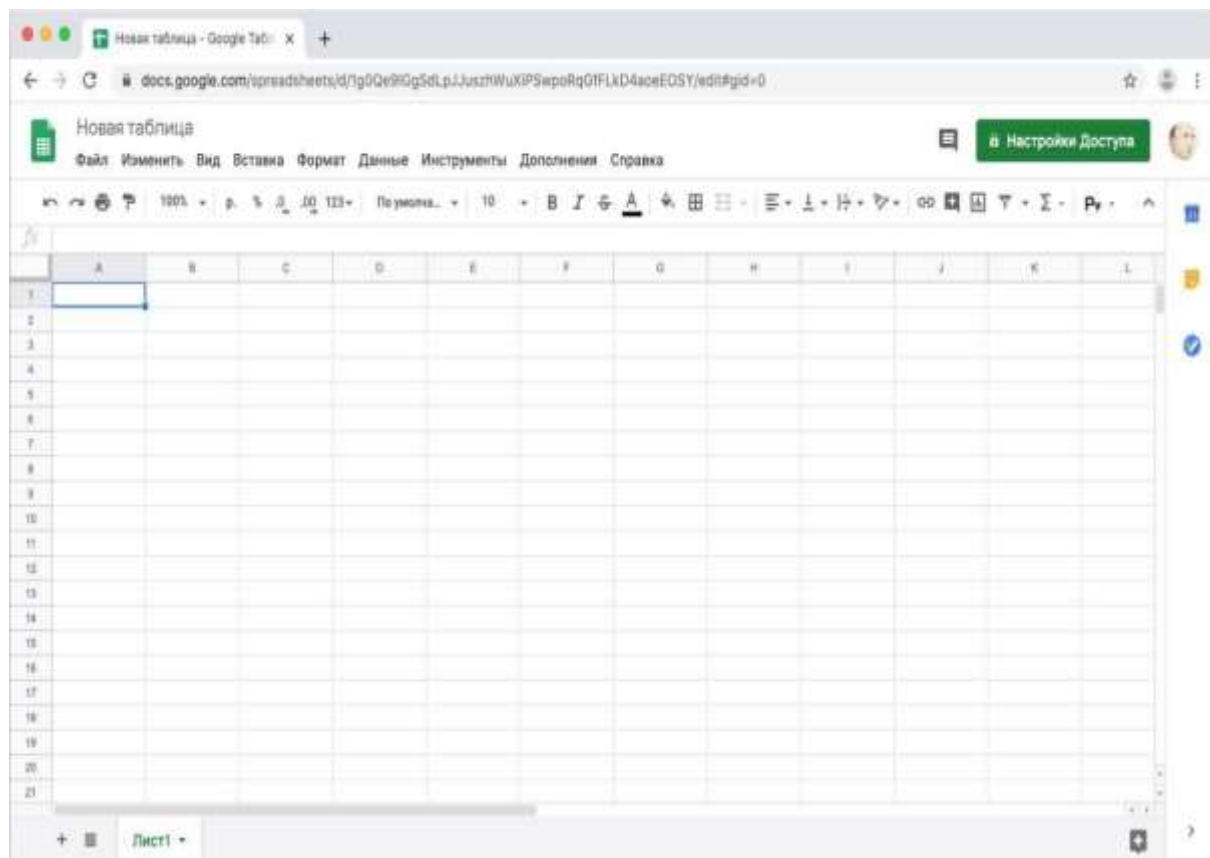


Рисунок 4 – Интерфейс веб–приложения Google Sheets

Однако, несмотря на все преимущества Google Sheets, для клиник особенно важно обеспечить высокий уровень безопасности при работе с конфиденциальной информацией пациентов. Использование собственных серверов

баз данных клиники может обеспечить дополнительные гарантии безопасности, такие как защита от несанкционированного доступа и возможность более тщательного контроля за хранением и обработкой данных. Это помогает предотвратить утечки информации и соблюсти законодательные требования по защите конфиденциальности пациентов.

LibreOffice Calc - это приложение, предназначенное для работы с электронными таблицами, которое входит в состав бесплатного офисного пакета LibreOffice, распространяемого с открытым исходным кодом. Оно предоставляет пользователю множество инструментов для создания, редактирования и форматирования таблиц, а также для выполнения различных расчетов и создания графиков.

Среди основных возможностей LibreOffice Calc можно выделить гибкое форматирование, позволяющее настраивать внешний вид таблицы в соответствии с требованиями пользователя. Также приложение поддерживает различные форматы файлов, что обеспечивает совместимость с другими офисными пакетами и удобство обмена данными.

Однако, несмотря на множество функций, анализ показывает, что интерфейс LibreOffice Calc может быть менее интуитивным и удобным для пользователей по сравнению с аналогичными приложениями. Это может создавать определенные трудности при освоении программы и повседневном использовании.

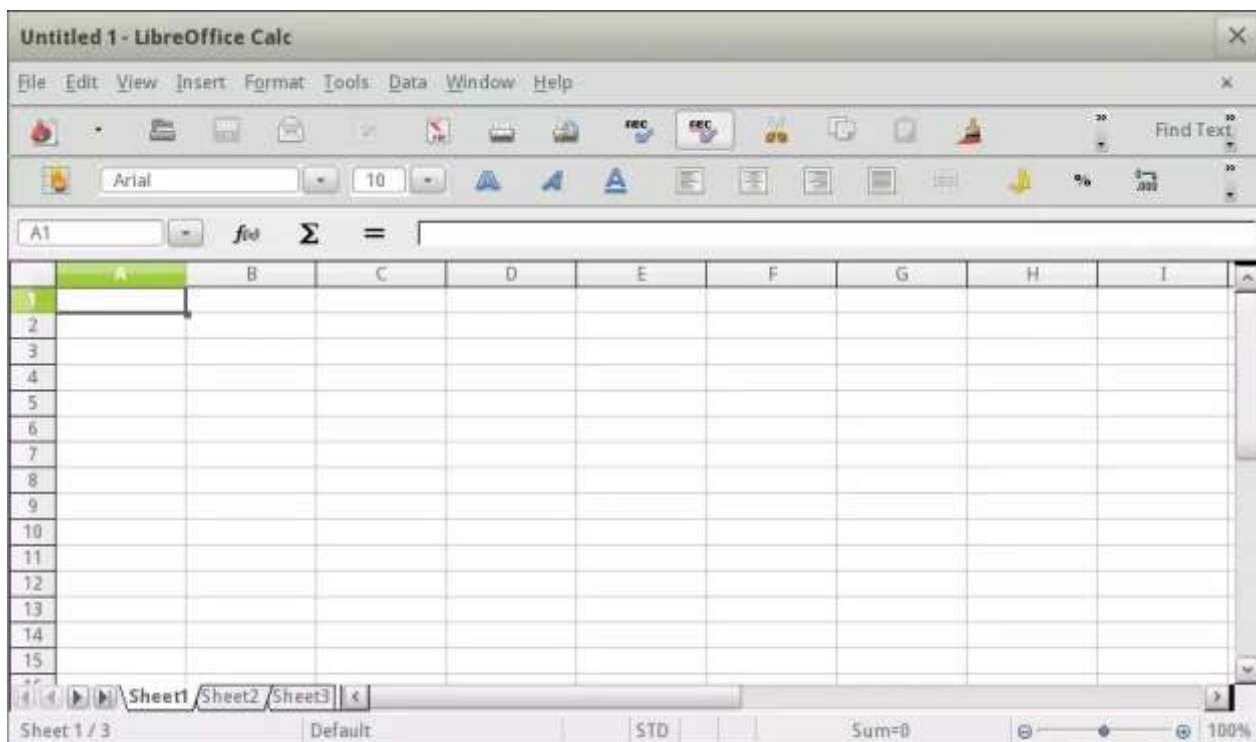


Рисунок 5 – Интерфейс программного средства LibreOffice Calc

Программное средство OpenOffice Calc – это приложение для работы с электронными таблицами, входящее в состав бесплатного офисного пакета OpenOffice. Данная программа позволяет пользователям создавать, редактировать и анализировать данные в табличной форме, а также строить графики и диаграммы.

OpenOffice Calc предлагает множество функций и инструментов для обработки данных, таких как автоматические расчеты, сортировка, фильтрация, использование формул и макросов.

В целом, OpenOffice Calc может быть полезным для тех, кто ищет бесплатное решение для работы с электронными таблицами, хотя оно во многом уступает другим программам в плане новейших технологий и функций.

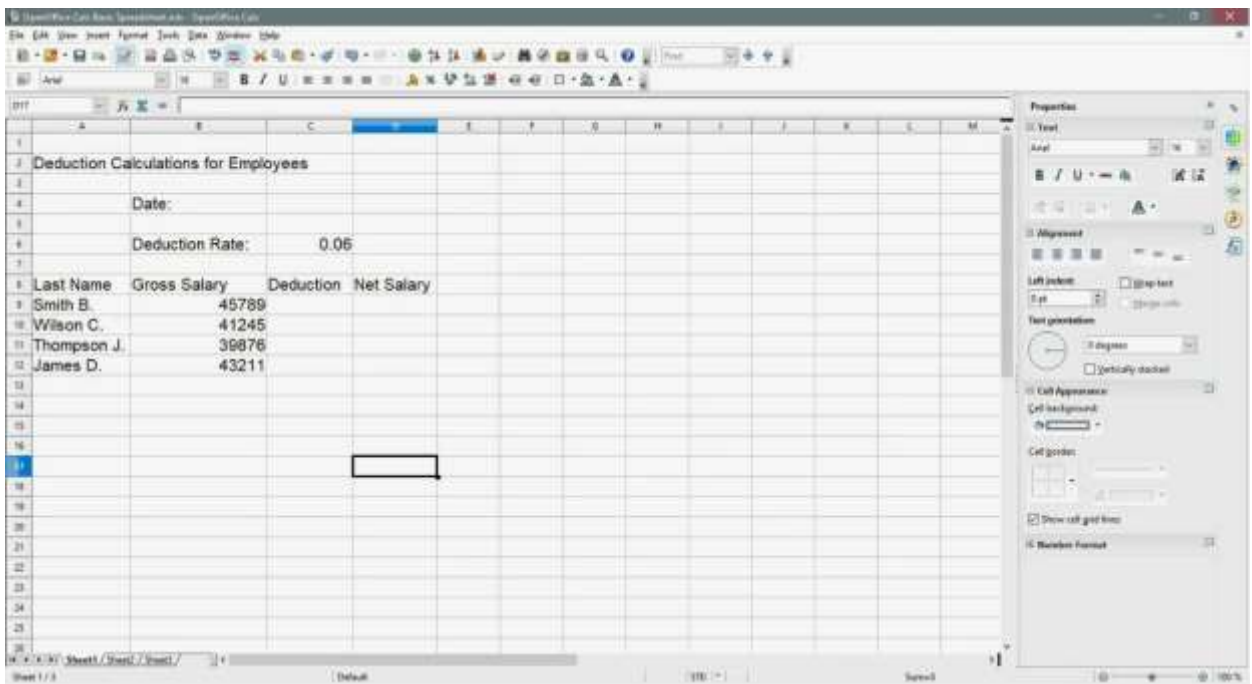


Рисунок 6 – Интерфейс программного средства OpenOffice Calc

Результат проанализированных аналогов представлен в таблице 1.

Таблица 1 – Анализ аналогов

Критерий оценки	Google Sheets	LibreOffice Calc	OpenOffice Calc
Гибкость и возможность настройки	–	–	–
Удобство использования	+	–	–
Возможность совместной работы	+	–	–
Бесплатное приложение	–	+	+

По итогам проведенного анализа стало очевидно, что ни одно из программных средств, рассмотренных в ходе исследования, не соответствует

специфическим требованиям и задачам стоматологической клиники «Зубная Архитектура». Каждый из протестированных программных продуктов имел свои достоинства, однако ни один из них не смог полностью удовлетворить все нужды клиники.

В связи с этим возникает необходимость разработки специализированного программного обеспечения, которое будет точно соответствовать требованиям клиники.

Разработка такого программного обеспечения обеспечит клинику «Зубная Архитектура» инструментами, необходимыми для эффективного и качественного обслуживания пациентов, а также оптимизации внутренних процессов.

1.7 Требования к функциональным особенностям

Для ведения учета были определены потребности, указанные в таблице.

Таблица 2 – Потребности предприятия

Потребность заказчика	Функциональные особенности
Табличное представление данных предприятия	Загрузка данных из базы в web–приложение; Представление таблиц в пользовательских формах.
Хранение записей о пациентах и врачах	Возможность вносить, редактировать и удалять записи.
Фильтрация записей	Возможности по осуществлению поиска по дате и ФИО врача или пациента.
Разграничение доступа для сотрудников и клиентов	Выполнение входа по логину и паролю; Выполнение авторизации; Вывод диалоговых окон об ошибках ввода логина и пароля; Разработать права доступа для обычных пользователей, обеспечивая для них отдельные функции приложения.

Для удовлетворения этих потребностей было предложено создания веб-приложения, включающего в себя несколько ключевых функций:

- веб-приложение должно предоставлять пользователям возможность авторизации и регистрации. Это означает, что пользователи смогут создать учетные записи, заполнив необходимые данные, такие как имя, адрес электронной почты и пароль;

- пользователи с соответствующим уровнем доступа веб-приложения получат возможность вносить, редактировать и удалять записи. Это позволит медицинским работникам легко добавлять новые записи, включая информацию о врачах, пациенте, а также об оказанных пациенту услугах. Функция редактирования позволит корректировать информацию в уже существующих записях, а возможность удаления обеспечит поддержание актуальности данных, исключая ненужные или ошибочные записи;

- возможность поиска по дате, имени врача и имени пациента. Это значительно облегчит врачам навигацию по системе и позволит быстро находить необходимые записи. Поиск по имени врача позволит быстро найти все приемы, связанные с конкретным специалистом, а поиск по имени пациента – получить всю информацию о визитах конкретного пациента;

- рассылка записей за определенный промежуток времени на выбранные адреса электронной почты. Веб-приложение будет генерировать письма, включающие данные о всех добавленных, отредактированных и удаленных записях за выбранный период;

- веб-приложение должно предоставлять клиентам возможность оставлять заявки на запись к врачу. Клиенты смогут указать предпочтительное время приема, указать номер для связи и кратко описать причину обращения. Эти заявки будут поступать в систему и обрабатываться администраторами или самими врачами, которые смогут подтвердить запись или предложить другое время приема. Это повысит удобство для пациентов и оптимизирует процесс записи на прием.

2 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1 Инфологическое проектирование базы данных

2.1.1 Сущности и их краткая характеристика

На основании анализа предметной области и структуры предприятия были определены следующие сущности, необходимые для создания информационной базы.

Все сущности приведены в таблице 3.

Таблица 3 – Формирование сущностей.

Название сущности	Описание сущности
Врачи	Содержит информацию о врачах клиники
Услуги	Содержит наименование всех услуг и их цену
Клиент	Содержит информацию о клиентах информационной системы

Далее представлены спецификации атрибутов сущностей:

Таблица 4 – Спецификация атрибутов сущности «Врачи»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
ID врача	Уникальный идентификатор пользователя	Числовой	> 0	1
ФИО	ФИО пользователя	Текстовый	–	Пляскин Алексей Сергеевич
Электронная почта	Адрес электронной почты пользователя	Текстовый	–	Plackin@mail.ru

Роль	Является ли пользователь администратором	Логический	0/1	0
Активен	Статус активности пользователя	Логический	0/1	1
Логин	Логин пользователя	Текстовый	–	Plasss
Пароль	Пароль пользователя	Текстовый	–	Dzfsdf555

Таблица 5 – Спецификация атрибутов сущности «Услуги»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
ID услуги	Уникальный идентификатор услуги	Числовой	> 0	1
Наименование услуги	Название услуги	Текстовый	–	Наложение временной пломбы
Цена	Стоимость услуги	Денежный	–	350

Таблица 6 – Спецификация атрибутов сущности «Клиент»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
-------------------	-------------------	------------	-------------------	-----------------

ID записи	Уникальный идентификатор записи	Числовой	> 0	1
ФИО	ФИО клиента	Текстовый	–	Багай Федор Петрович
Сумма	Сумма всех оказанных услуг	Денежный	–	3500
Дата	Дата записи	Дата и время	Текущая дата записи	2024-05-22 00:17:19.433770

2.1.2 Описание связей между сущностями

Установление связи между сущностями «Клиент» и «Врач» описаны на рисунке 7. В этом случае имеется связь «один ко многим». У одного врача может быть несколько клиентов. Это означает, что врач сможет заносить, редактировать и удалять записи всех своих пациентов.



Рисунок 7 – Связь «Врач – Клиент»

Установление связи между сущностями «Клиент» и «Услуги» описаны на рисунке 8. В этом случае имеется связь «многие ко многим». У одного клиента, может быть, несколько услуг и одна услуга может быть оказана несколькими клиентам.

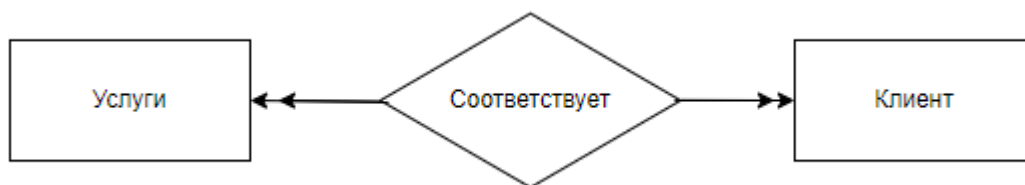


Рисунок 8 – Связь «Услуги – Клиент»

Чтобы предотвратить ошибки, была создана специальная сущность «КлиентУслуга». Она делает возможным преобразование связи «многие ко многим» в две связи «один ко многим». Данные связи представлены на рисунке 9.

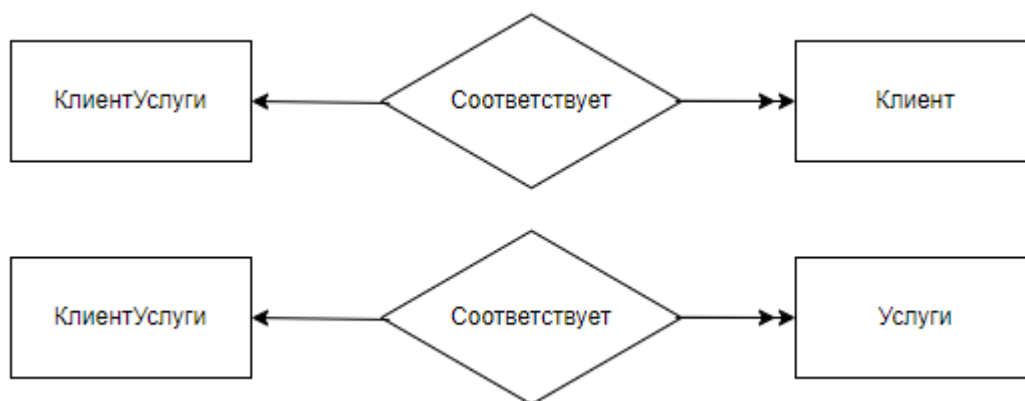


Рисунок 9 – Преобразование связи «многие ко многим» в «один ко многим»

Инфологическая модель:

Врач

<u>ИД Пользователя</u>	ФИО	Электронная почта	Роль	Активность	Логин	Пароль
------------------------	-----	-------------------	------	------------	-------	--------

Услуги

<u>ИД Услуги</u>	Наименование услуги	Цена
------------------	---------------------	------

Клиент

<u>ID записи</u>	ФИО	Сумма	Дата
------------------	-----	-------	------

2.1.3 Нормализация отношений

Давайте проверим, соответствуют ли полученные отношения трем стандартным формам.

– преобразование отношений к первой стандартной форме. Все отношения, полученные из концептуальной информационной модели данных в реляционную форму, являются атомарными, то есть каждое значение атрибута не представляет собой набор или повторяющуюся группу. Следовательно, все отношения находятся в первой стандартной форме;

– преобразование отношений ко второй стандартной форме. Отношение находится во второй стандартной форме, если оно уже находится в первой стандартной форме, и каждый не ключевой атрибут функционально зависит от ключа. Следовательно, все отношения находятся во второй стандартной форме;

– преобразование отношений к третьей стандартной форме. Отношение находится во второй стандартной форме и не имеет транзитивных зависимостей между не ключевыми атрибутами.

2.2 Логическое и физическое проектирование базы данных

Логическое проектирование – это процесс разработки информационной модели, которая не зависит от выбранной системы управления базами данных и других условий физической реализации.

На основе логической модели создается физическая модель базы данных. Процесс проектирования структуры данных включает создание физической структуры таблиц для каждого отношения.

Благодаря встроенным по умолчанию возможностям Django, после инфологического проектирования в моделях, логическое и физическое проектирование выстраивается автоматически.

Посмотреть Логическую и физическую модель можно с помощью стороннего приложения отображения базы данных DBeaver.

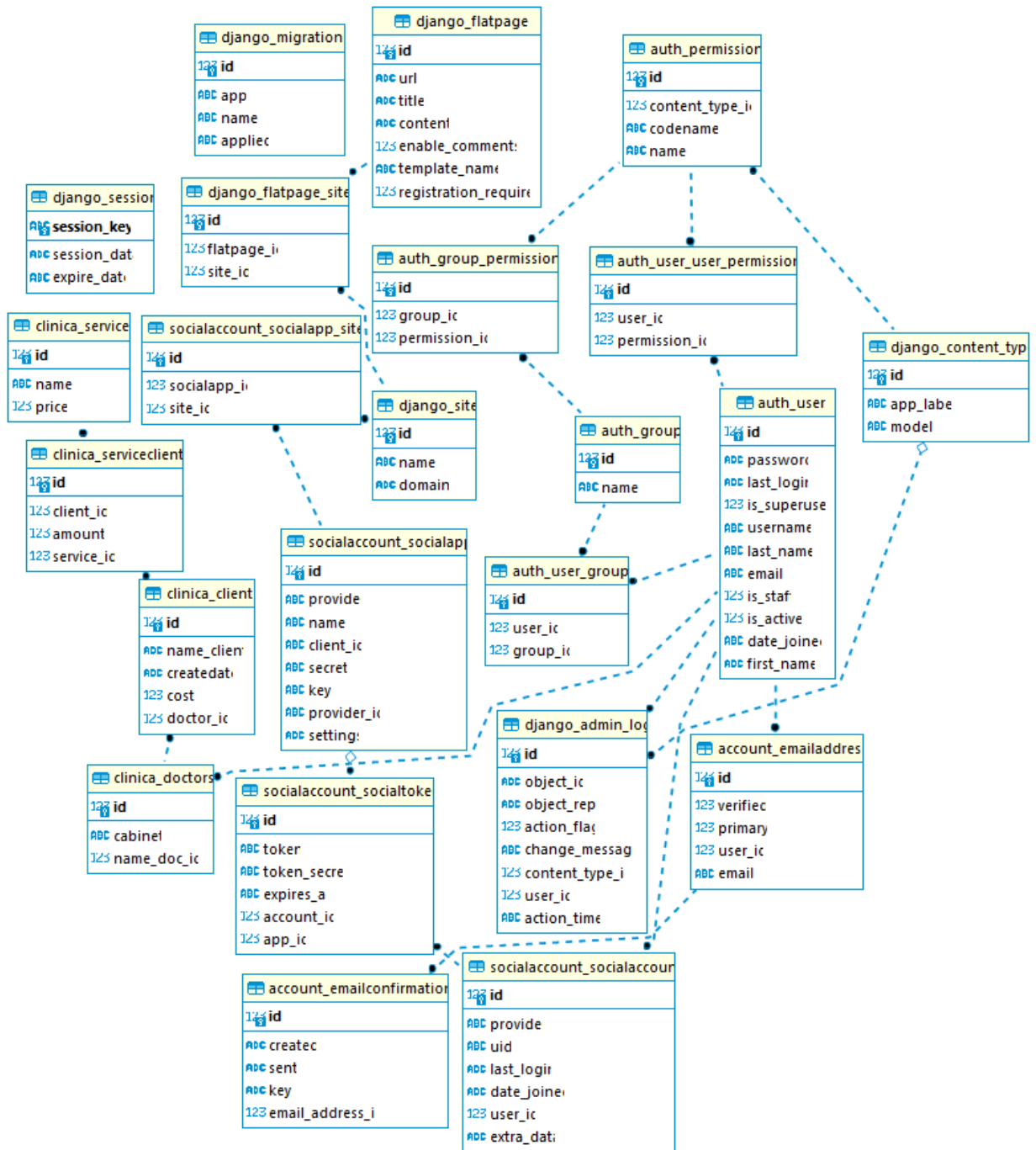


Рисунок 10 – Логическая модель базы данных

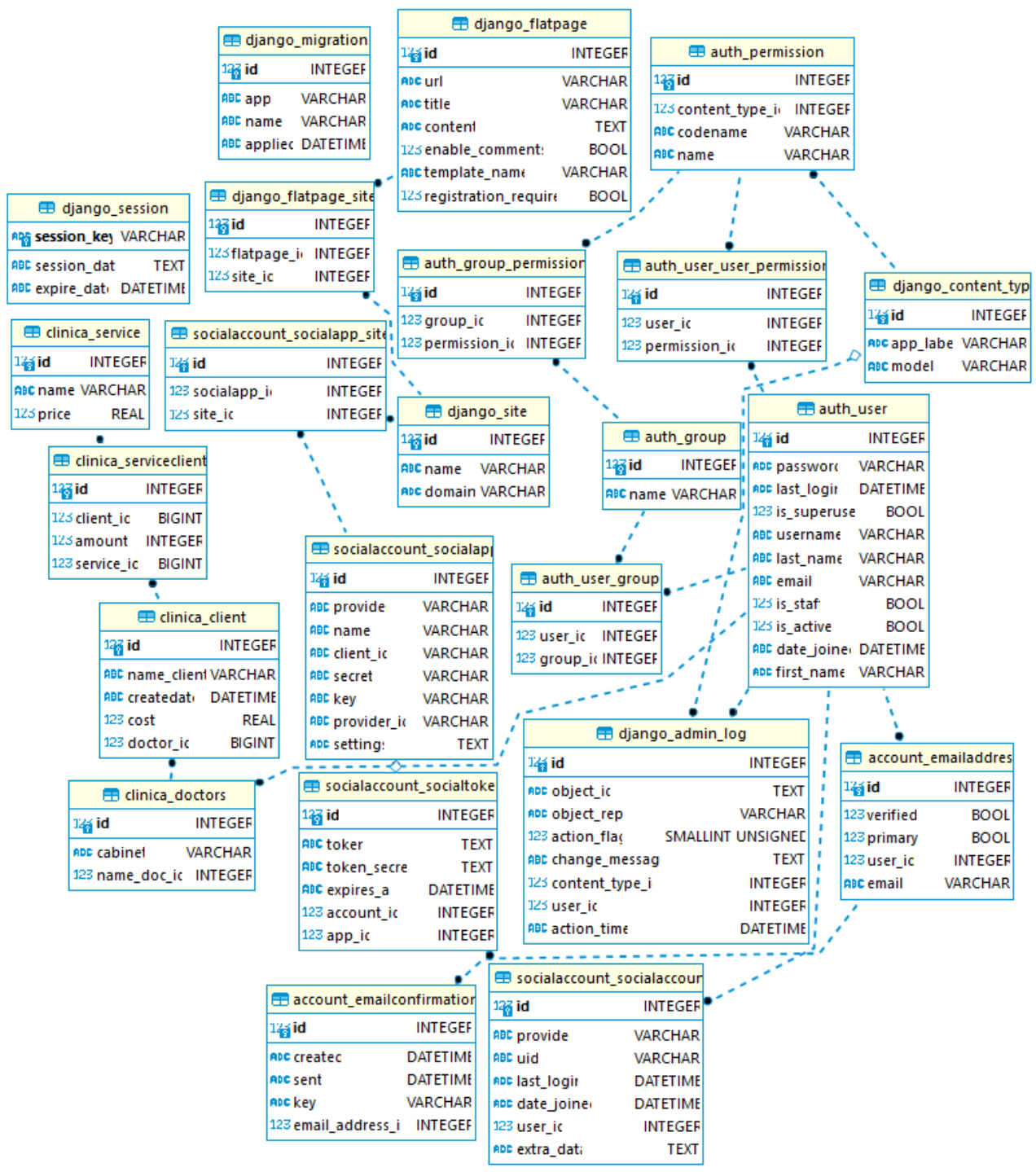


Рисунок 11 – Физическая модель базы данных

3 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

3.1 Назначение и цели создания

Создаваемая автоматизированная платформа представляет собой инструмент для оптимизации финансовой учетной деятельности медицинского учреждения, обеспечивая запись и обработку информации об оказанных услугах, а также осуществление запросов.

Основной целью разработки такой системы является повышение эффективности функционирования клиники, усовершенствование учета информации, а также обеспечение надежности и целостности данных.

В рамках этих целей автоматизированная система должна выполнять следующие задачи:

- возможность авторизации и регистрации;
- возможность заносить\редактировать\удалять записи;
- возможность поиска по дате\имени врача\имени пациента;
- возможность клиентам оставлять заявки на запись;
- рассылка отчетов о занесенных записях на указанные почты;
- кеширование данных;
- отслеживание ошибок.

3.2 Разработка технического задания

Исходя из требований заказчика к системе было составлено техническое задание (Приложение А).

ТЗ необходимо для того, чтобы определить полный порядок действий по созданию, разработке и внедрению на информационную площадку программного продукта и то, что из себя он должен представлять.

Полное наименование: Разработка финансового веб-приложения для стоматологической клиники ООО «Зубная архитектура».

Разработчик: Студент группы 0103–об, института компьютерных и инженерных наук, кафедры информационных и управляющих систем, Амурского государственного университета Беднова Дарья Сергеевна.

Заказчик: Общество с ограниченной ответственностью ООО «Арх-ДЕНТ», Быков Илья Павлович.

Юридический адрес: 675000, Амурская область, г. Благовещенск, Игнатьевское шоссе, 15 (2-й мкр.).

Подпись заказчика и исполнителя в данном документе подтверждает согласие обеих сторон со следующими условиями:

- исполнитель разрабатывает техническое задание;
- заказчик согласен со всеми пунктами, которые и излагаются в техническом задании;
- исполнитель обязуется исполнить полный объем работы, указанные в настоящем техническом задании в обусловленные заранее сроки;
- заказчик вправе требовать от исполнителя только те виды работ или услуг, которые точно описаны в техническом задании;
- все неточности требующие исправления или доработки, выявленные в настоящем документе после его подписания, подлежат двухстороннему согласованию между исполнителем и заказчиком.

Данная работа сделана в учебных целях и осуществляется без привлечения каких–либо финансовых ресурсов.

3.3 Функциональные требования

Разработка автоматизированной системы для стоматологической клиники «Зубная Архитектура» на основе представленных целей и задач должна соответствовать следующим функциональным требованиям:

Критерии со стороны пациента:

- возможность регистрироваться в веб приложении;
- возможность оставить заявку для записи на прием.

Критерии со стороны врача:

- возможность записывать пациента и оказанные ему услуги с выводом итоговой суммы;
- возможность поиска нужных записей по дате, ФИО врача или пациента;
- возможность получать письма на почту с записями за определенный период.

Критерии со стороны администратора:

- возможность просмотра всех таблиц;
- отображение списка активных пользователей системы с указанием их статуса;
- возможность добавлять новых пользователей, устанавливать им права доступа;
- просмотр списка всех текущих пользователей системы с детальной информацией о них, их ролях и правах доступа для управления пользователями;
- возможность настроить периодичность отправки писем с записями на почту.

3.4 Техническое обеспечение

При выборе языка программирования, Python выделяется своей простотой в освоении и эффективностью в использовании. Именно эти качества привлекли внимание и стали определяющими при выборе языка для разработки проекта. Помимо этого, Python обладает активным сообществом разработчиков, что гарантирует быструю поддержку и решение любых возникающих проблем.

Выбор среды разработки также играет ключевую роль в процессе создания программного обеспечения. PyCharm, разработанный компанией JetBrains, заслуживает доверия благодаря своей надежности и многочисленным функциям, которые значительно упрощают процесс разработки. Его удобный интерфейс, интеграция с различными инструментами и возмож-

ность расширения функциональности делают его одним из предпочтительных выборов для разработчиков Python.

Django, в свою очередь, представляет собой не просто фреймворк, а полноценный инструмент для создания веб-приложений. Его мощные функции позволяют разрабатывать высокопроизводительные и масштабируемые приложения, при этом соблюдая лучшие практики разработки программного обеспечения. Благодаря Django, разработка веб-приложений на Python становится быстрой и эффективной, а архитектура приложения остается легко поддерживаемой и расширяемой.

Интеграция PyCharm с Django дополняет этот набор преимуществ, обеспечивая комфортную среду разработки, специально адаптированную для работы с этим фреймворком. Разработчики получают доступ к широкому набору инструментов, включая отладку, анализ кода, автодополнение и многое другое, что существенно ускоряет процесс создания приложений и повышает их качество.

Таким образом, комбинация Python, Django и PyCharm представляет собой мощный инструментарий для разработки программного обеспечения любой сложности. Она сочетает в себе простоту, эффективность и надежность, делая процесс разработки приложений приятным и продуктивным.

Первоочередная задача при начале разработки приложения на Python в среде PyCharm — это установка виртуальной среды. Виртуальная среда играет ключевую роль, предотвращая наложение пакетов и конфликты версий между приложениями, что помогает обеспечить стабильную работу разрабатываемого приложения.

После установки виртуальной среды следует приступить к установке Django и проверке правильности его установки путем запуска программы.

Окно по умолчанию Django должно выглядеть так, как на рисунке 3. Если отображение отличается от ожидаемого, следует провести дополнительные проверки настройки проекта и установки Django.

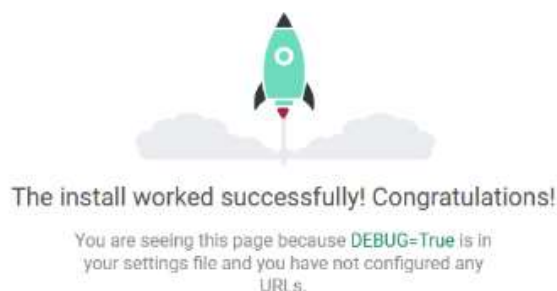


Рисунок 12 – Окно программы по умолчанию.

После установки Django в проекте автоматически создается файл настроек, который обычно называется `settings.py`. Этот файл содержит конфигурационные параметры для вашего Django-приложения, такие как настройки базы данных, путей к статическим файлам, настройки безопасности и многое другое.

Чтобы файл настроек корректно работал, необходимо провести миграцию. Миграция — это процесс, при котором Django применяет изменения в моделях данных вашего приложения к базе данных. Это важно, потому что файл настроек может содержать ссылки на модели данных, а без миграции изменения в моделях не будут отражены в базе данных.

В дальнейшем, при добавлении новой информации в файл настроек, также необходимо будет проводить миграции. Это связано с тем, что измене-

ния в настройках могут затронуть структуру базы данных или другие аспекты, которые требуют обновления существующих данных или структур.

При добавлении новых параметров в файл настроек Django, рекомендуется следовать принципу "сначала миграция, потом применение изменений". То есть сначала добавить необходимые изменения в модели данных приложения, затем провести миграцию и только после этого внести изменения в файл настроек и провести новую миграцию, если это необходимо.

3.5 Работа с Django на Python

3.5.1 Написание моделей и представлений

В Django, модели (Models) и представления (Views) представляют собой два ключевых компонента архитектуры MVC (Model-View-Controller) или ее вариации MTV (Model-Template-View), которые используются для разработки веб-приложений.

Модели (Models):

Модели в Django определяют структуру данных приложения и методы взаимодействия с этими данными. Каждая модель представляет собой отдельный тип данных, такой как пользователь, пост в блоге, комментарий и так далее. Они создаются с использованием классов Python, которые наследуются от `django.db.models.Model`. Внутри каждого класса модели определяются атрибуты, соответствующие полям в базе данных, а также методы, которые определяют поведение модели.

Поля модели могут быть различных типов: строки, числа, даты, булевы значения и другие, а также специальные поля, такие как `ForeignKey`, `ManyToManyField` и `OneToOneField`, которые используются для описания связей между моделями.

Кроме того, модели могут включать методы для выполнения различных операций с данными, таких как сохранение, обновление, удаление и запросы к базе данных. Эти методы могут быть определены непосредственно в классе модели или добавлены с использованием менеджеров объектов

(objects managers).

В конечном итоге, модели Django обеспечивают удобное и эффективное средство для работы с данными в веб-приложениях, обеспечивая их структурирование, связывание и обработку.

```
class Client(models.Model):
    name_client = models.CharField(max_length=255)
    createdate = models.DateTimeField(auto_now_add=True)
    serv = models.ManyToManyField(Service, through='ServiceClient')
    cost = models.FloatField(default=0.0)
    doctor = models.ForeignKey(Doctors, on_delete=models.CASCADE)

    def __str__(self):
        return f"{self.cost}"

class ServiceClient(models.Model):
    service = models.ForeignKey(Service, on_delete=models.CASCADE)
    client = models.ForeignKey(Client, on_delete=models.CASCADE)
    _amount = models.IntegerField(default=1, db_column='amount')

    def service_sum(self):
        return self.service.price * self.amount
```

Рисунок 13 – Модели проекта

Представления (Views):

Представления в Django отвечают за обработку запросов и формирование ответов. Они используют данные из моделей для формирования ответа, который может быть в виде HTML для отображения в браузере или JSON для API. Представления создаются в виде функций или классов Python. В функциональных представлениях вам нужно явно обработать запрос и вернуть ответ. В классах представлений вы обычно переопределяете методы, такие как `get()` или `post()`, для обработки соответствующих методов.

Кроме того, представления могут быть организованы с использованием

классов-наследников Django's generic views, что позволяет сократить количество кода за счет использования готовых решений и шаблонов поведения. Классы-представления предоставляют гибкую структуру для обработки различных типов запросов и позволяют более легко организовать код благодаря использованию наследования и методов, специализированных для различных типов HTTP-запросов.

```
class ClientAdd(CreateView):
    model = Client
    template_name = 'add.html'
    context_object_name = 'add'
    queryset = Client.objects.order_by('-createdate')
    paginate_by = 10
    form_class = ClientForm # добавляем форм класс, чтобы получ
    success_url = reverse_lazy('main')

class ClientSearch(ListView):
    model = Client
    template_name = 'search.html'
    context_object_name = 'search'
    queryset = Client.objects.order_by('-createdate')
    paginate_by = 10

class ClientUpdateView(UpdateView):
    template_name = 'add.html'
    form_class = ClientForm
    success_url = reverse_lazy('doclist')
```

Рисунок 14 – Представления проекта

Представления веб-приложений играют важную роль в получении, обновлении и сохранении данных. Они действуют как промежуточное звено между пользовательским запросом и базой данных. Обычно в представлении

ях используются методы моделей Django для выполнения запросов к базе данных и получения необходимых данных для формирования ответа на запросы. Важно отметить, что модели не имеют непосредственного отношения к представлениям; они занимаются исключительно работой с данными.

Представления в Django обычно взаимодействуют с шаблонами (Templates), которые используют данные, предоставленные представлением, для формирования конечного HTML-кода или других форматов ответа, который отправляется пользователю.

Шаблоны предоставляют гибкий способ организации и структурирования пользовательского интерфейса, позволяя легко встраивать данные, полученные из представлений, в HTML-код. Таким образом, представления и шаблоны совместно обеспечивают создание динамических и интерактивных веб-приложений на основе данных из базы данных.

3.5.2 Написание шаблонов

В Django, URL (Uniform Resource Locator) играют важную роль в направлении пользователей к определенным частям вашего веб-приложения. Они определяют, какие представления (views) должны быть вызваны для конкретных запросов URL.

Вот как это работает подробнее:

- URL-шаблоны (URL patterns): В Django для маршрутизации URL используются URL-шаблоны. Это правила, которые определяют, какие URL-адреса будут обрабатываться какими представлениями. URL-шаблоны задаются в файле `urls.py` в каждом приложении Django.

- файл `urls.py`: В Django каждое приложение имеет свой файл `urls.py`, который определяет, какие URL маршруты должны обрабатываться какими представлениями. Обычно это выглядит как список URL-шаблонов вида `urlpatterns = [...]`;

- функции представлений (views): URL-шаблоны направляют запросы к функциям представлений. Функции представлений принимают за-

прос, обрабатывают его и возвращают ответ. Они могут возвращать HTML-страницы, данные в формате JSON и так далее;

```
from django.urls import path
from .views import DoctorsList, Servicelist, DoctorsDetail, ClientSearch, ClientAdd,

urlpatterns = [
    path('', DoctorsList.as_view(), name='main'),
    path('<int:pk>/', DoctorsDetail.as_view(), name='doc_detail'),
    path('cli/', ClientList.as_view(), name='cli_list'),
    path('<int:pk>/clients/', ClientDetail.as_view(), name='cli_detail'),
    path('search/', ClientSearch.as_view(), name='cli_search'),
    path('add/', ClientAdd.as_view(), name='cli_add'),
    path('update/<int:pk>/', ClientUpdateView.as_view()),
    path('delete/<int:pk>/', ClientDeleteView.as_view(), name='delite'),
    path('cost/', Servicelist.as_view(), name='cost'),
]
```

Рисунок 15 – urls.py файл

- именованные URL (Named URL): Вы можете называть URL-шаблоны для более легкого обращения к ним в шаблонах и в других частях приложения. В приведенном выше примере, у URL-шаблона index есть имя 'index', что позволяет ссылаться на него в шаблонах и коде с помощью `reverse('index')`;

- передача параметров в URL: Вы можете передавать параметры в URL, чтобы определить уникальные части URL-адресов. В примере выше `<int:article_id>` указывает на то, что в этом месте ожидается целое число, которое будет передано в функцию представления `article_detail` как `article_id`;

- в Django также можно использовать `include`, чтобы разделить определения URL-шаблонов на более мелкие части и включить их в основной файл `urls.py`;

URL-адреса являются ключевой частью любого веб-приложения Django, поскольку они определяют структуру и навигацию в вашем прило-

жении. Правильное использование URL-шаблонов позволяет эффективно управлять запросами и обеспечивать удобство использования вашего веб-приложения пользователями.

При использовании платформы Django, интеграция библиотеки шаблонов Bootstrap становится критически важной для создания стильного и современного пользовательского интерфейса. Bootstrap предлагает обширный набор готовых компонентов, таких как модальные окна, формы, кнопки, навигационные панели и многое другое, обеспечивая единообразный дизайн и поведение элементов интерфейса.

Использование Bootstrap в Django значительно ускоряет процесс разработки, поскольку множество компонентов уже реализованы и не требуют дополнительного кода. Кроме того, благодаря адаптивной верстке Bootstrap приложения могут корректно отображаться на различных устройствах и экранах.

После установки библиотеки необходимо создать файл, который будет отображать дизайн по умолчанию. Обычно скачанный файл шаблона сохраняется под именем `default.html`, и он служит основным шаблоном, от которого происходит загрузка дизайна во все остальные файлы.

Интеграция Bootstrap в Django существенно улучшает пользовательский опыт, обеспечивая современный и удобный интерфейс при минимальных затратах времени и ресурсов на разработку. Благодаря широкому выбору компонентов и простоте использования, Bootstrap остается одним из наиболее популярных инструментов для создания веб-приложений на основе Django.

```

<!DOCTYPE html>
{% load static %}
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>Зубная Архитектура</title>
  <meta content="" name="description">
  <meta content="" name="keywords">

  <!-- Favicons -->
  <link href="{% static 'assets/img/favicon.png' %}" rel="icon">
  <link href="{% static 'assets/img/apple-touch-icon.png' %}" rel="apple-touch-icon">

  <!-- Google Fonts -->
  <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i" rel="stylesheet">

  <!-- Vendor CSS Files -->
  <link href="{% static 'assets/vendor/animate.css/animate.min.css' %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/bootstrap/css/bootstrap.min.css' %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/bootstrap-icons/bootstrap-icons.css' %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/boxicons/css/boxicons.min.css' %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/glightbox/css/glightbox.min.css' %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/swiper/swiper-bundle.min.css' %}" rel="stylesheet">

```

Рисунок 16 – default.html файл

Связь между дизайном пользовательского интерфейса и функциональным кодом приложения играет ключевую роль в обеспечении эффективной работы программного продукта. Особенно это актуально в контексте веб-разработки, где важно не только обеспечить пользователей привлекательным и удобным интерфейсом, но и грамотно организовать логику обработки данных.

Представления, используемые в фреймворке Django, выступают важным звеном, обеспечивающим связь между дизайном страниц и функциональным кодом приложения. Они действуют как посредники, преобразуя запросы пользователя в соответствующие действия и обратно, что позволяет приложению эффективно взаимодействовать с пользователем.

Каждое представление в Django связывается с определенной функцией

или методом, которые отвечают за обработку запросов и возврат соответствующих ответов. Такой подход позволяет приложению получать данные, введенные пользователем, обрабатывать их в соответствии с бизнес-логикой приложения и отображать результаты на экране в удобном для пользователя виде.

Эффективное взаимодействие между дизайном интерфейса и функциональным кодом обеспечивает пользовательский опыт высокого качества, увеличивает удовлетворенность пользователей и повышает конкурентоспособность веб-приложения на рынке. Важно, чтобы дизайн и функциональность дополняли друг друга, создавая цельное и эффективное программное решение.

В результате, благодаря этой взаимосвязи, приложение может эффективно функционировать, обеспечивая пользователям удобный и интуитивно понятный интерфейс для взаимодействия с данными.

3.5.3 Кэширование и отслеживания ошибок

В целях обеспечения безопасности и контроля за работой веб-приложения было внедрено кэширование данных и запись об ошибках.

Кэширование данных позволяет ускорить доступ к часто используемым ресурсам, таким как статические файлы или результаты запросов к базе данных. Это улучшает производительность приложения и снижает нагрузку на сервер.

Кроме того, кэширование может повысить надежность и отказоустойчивость приложения. Если сервер временно недоступен или возникают проблемы с сетью, кэшированные данные могут быть использованы для обеспечения продолжительности работы приложения и сохранения пользовательского опыта.

Запись об ошибках также является важной составляющей системы контроля качества приложения. Когда возникают ошибки или сбои в работе приложения, эти записи фиксируют информацию о проблеме, включая тип

ошибки, место возникновения и дополнительные данные, которые могут помочь в выявлении и устранении проблемы.

Общая система контроля и обработки ошибок помогает обеспечить стабильную и надежную работу приложения, а также быстро реагировать на любые проблемы, которые могут возникнуть в процессе его использования.

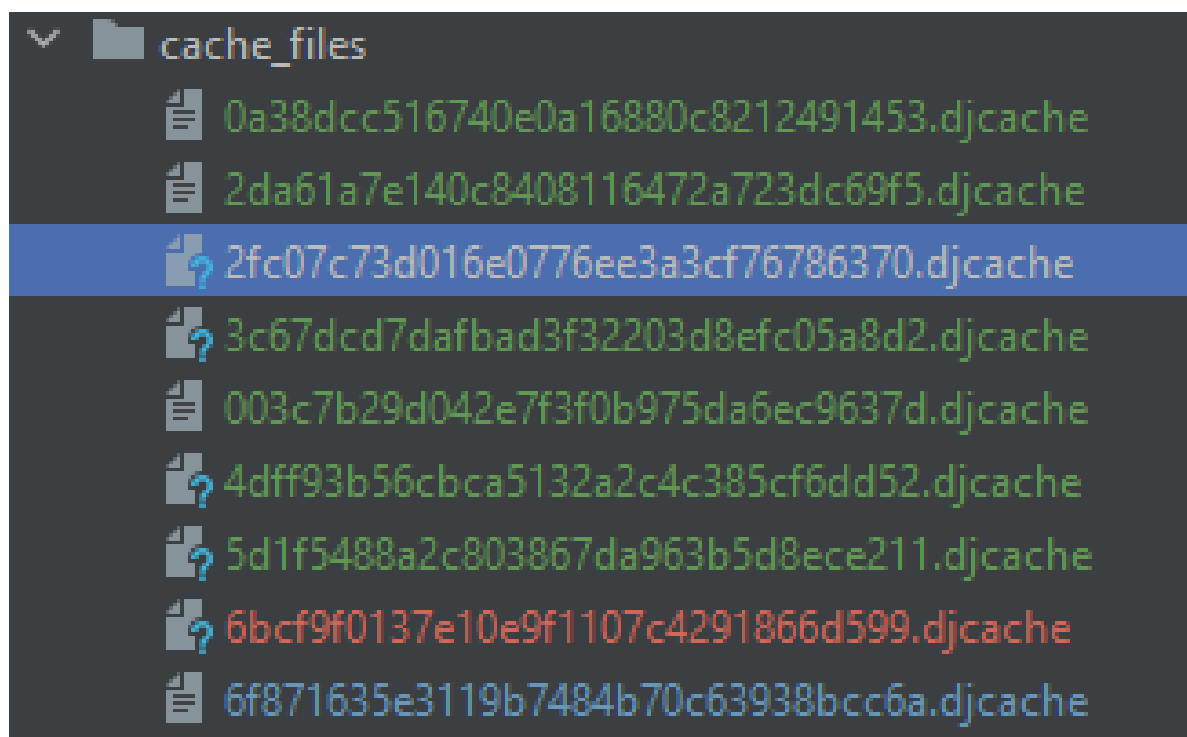
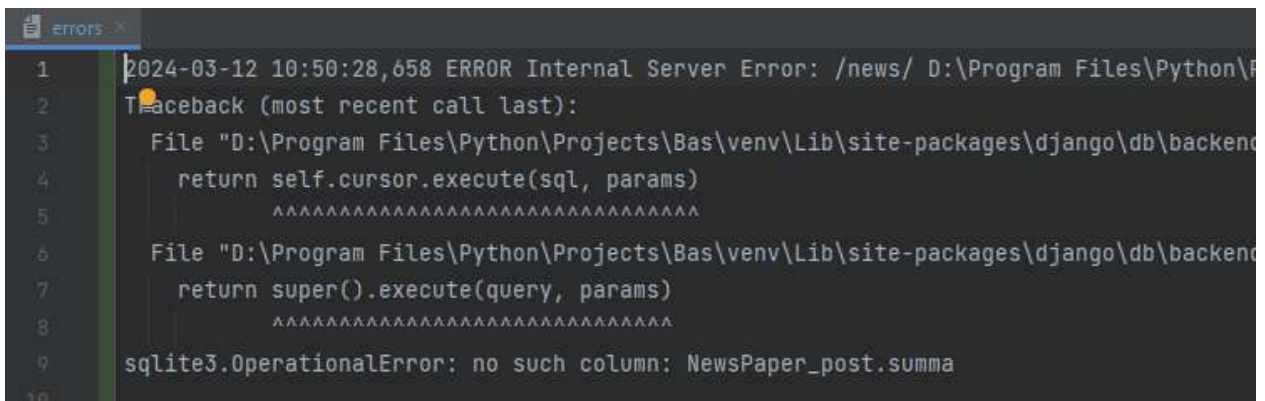


Рисунок 17 – Кеш файлы

Кэширование является одним из основных способов снизить нагрузку на веб-приложение при одновременно выполняемых запросов от разных пользователей.



```
errors x
1 | 2024-03-12 10:50:28,658 ERROR Internal Server Error: /news/ D:\Program Files\Python\F
2 | Traceback (most recent call last):
3 |   File "D:\Program Files\Python\Projects\Bas\venv\Lib\site-packages\django\db\backen
4 |     return self.cursor.execute(sql, params)
5 |     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
6 |   File "D:\Program Files\Python\Projects\Bas\venv\Lib\site-packages\django\db\backen
7 |     return super().execute(query, params)
8 |     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
9 | sqlite3.OperationalError: no such column: NewsPaper_post.summa
10 |
```

Рисунок 18 – Записи об ошибках

Файл об ошибках помогает администратору и разработчику понять, если ли ошибки и в какой части кода они возникли.

3.6 Работа с запросами и рассылками

3.6.1 Создание записи на прием

Вопрос об отправке электронных писем настолько часто встречается в любых веб-приложениях, что во многих системах уже есть готовые решения. Django, конечно же, не исключение, и в нём есть модуль `django.core.mail`.

`django.core.mail` - это встроенный модуль Django, который предоставляет простой интерфейс для отправки электронных писем из ваших приложений Django. Он упрощает отправку электронных писем, обеспечивая стандартизированный и удобный способ взаимодействия с SMTP-серверами.

Использование такого способа для отправки писем имеет ряд преимуществ `django.core.mail`:

- интеграция с Django: Поскольку `django.core.mail` является частью Django, его использование обеспечивает хорошую интеграцию с остальными компонентами фреймворка. Это делает его предпочтительным выбором для отправки электронной почты в приложениях Django;
- простота использования: API `django.core.mail` довольно прост в использовании. Он предоставляет функции, такие как `send_mail()`, которые позволяют отправлять электронные письма всего одной строкой кода;
- конфигурируемость: Модуль `django.core.mail` легко настраивается с помощью параметров в файле `settings.py` вашего проекта. Вы можете легко

настроить SMTP-сервер и другие параметры, такие как порт и протокол без необходимости вручную настраивать подключение к SMTP-серверу в коде;

- поддержка различных бэкендов: Помимо поддержки SMTP, `django.core.mail` также предоставляет возможность использовать различные бэкенды для отправки электронных писем. Это позволяет легко переключаться между бэкендами в зависимости от ваших потребностей, например, для тестирования;

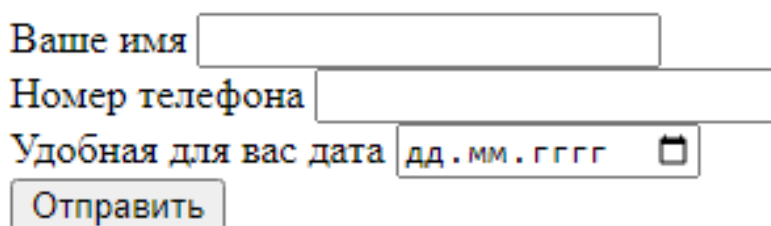
- управление ошибками: Модуль автоматически обрабатывает ошибки, связанные с отправкой писем. Вы можете настроить поведение в случае ошибок, чтобы ваше приложение могло адекватно реагировать на них, например, путем логирования или отправки уведомлений администраторам;

В целом, использование `django.core.mail` делает отправку электронных писем в приложениях Django простой и надежной задачей.

Именно этот способ был использован для добавления возможности отправлять клиентам заявки для записи на прием.

В форму заявки клиент вводит свое ФИО, номер телефона для связи и дату, на которую клиент хочет записаться.

Заявка на прием к врачу



Ваше имя

Номер телефона


Удобная для вас дата 

Рисунок 19 – Форма заявки

После отправление клиентом заявки, вся необходимая информация поступает на почту администратора, чтобы связаться с клиентом и уточнить

более подробно проблему и удобную дату для клиента.

Клиенту же, после отправки заявки придет на указанную почту при регистрации письмо, где клиента уведомят об оставлении заявки.

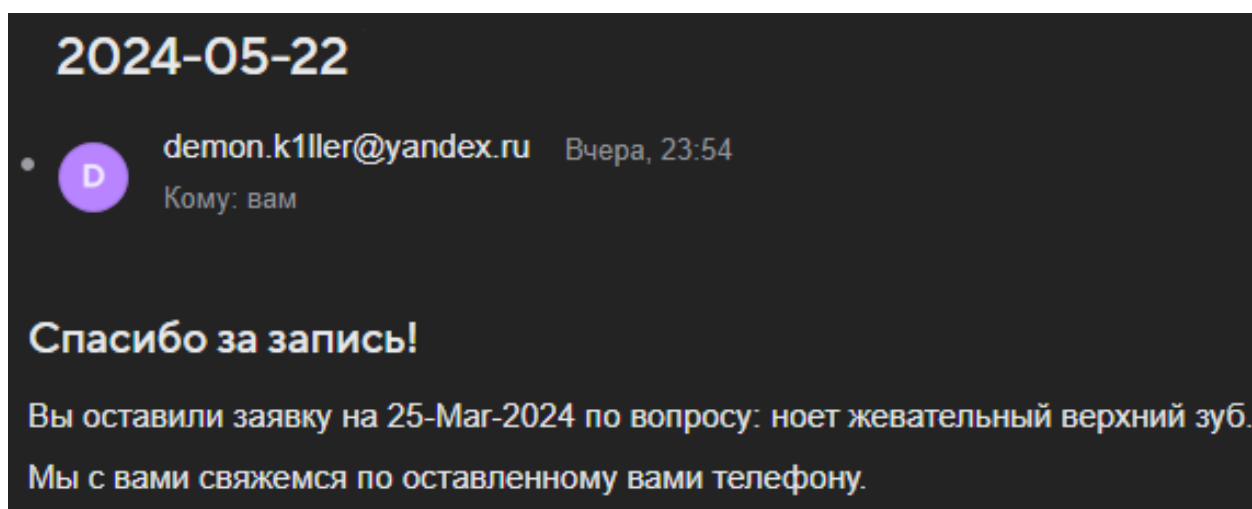


Рисунок 20 – Уведомление клиенту

3.6.2 Создание рассылки по расписанию

Когда речь идет об периодических отправках, то `django.core.mail` становится бессильным. В таком случае, для разработчика на Django приходят на помощь сигналы.

В Django сигналы — это механизм, который позволяет отправлять и принимать уведомления о событиях, происходящих в приложениях. Сигналы используются для связывания функций, которые будут выполняться в ответ на определенные события, например, создание, обновление или удаление объектов модели.

Основная идея заключается в том, что можно зарегистрировать обработчики сигналов, которые будут автоматически вызываться, когда происходит определенное событие. Это помогает разделить логику приложения на более мелкие и автономные части, что делает код более поддерживаемым и понятным.

Сигналы в Django особенно полезны в следующих случаях:

- автоматические обновления: возможно использовать сигналы для

автоматического выполнения определенных действий при изменении данных в базе данных, таких как отправка уведомлений или обновление связанных объектов;

– логирование и аудит: с помощью сигналов можно отслеживать и регистрировать различные события, происходящие в приложении, что помогает в анализе работы приложения и выявлении проблем;

– расширение функциональности: можно использовать сигналы для добавления дополнительной функциональности к сторонним приложениям, не изменяя их исходного кода;

– синхронизация данных: с помощью сигналов можно обеспечить синхронизацию данных между различными частями приложения или сервисами.

Данный случай соответствует первому.

Сигналы обеспечивают отставку списка записей за прошедшую неделю. Пример представлен на рисунке 21.

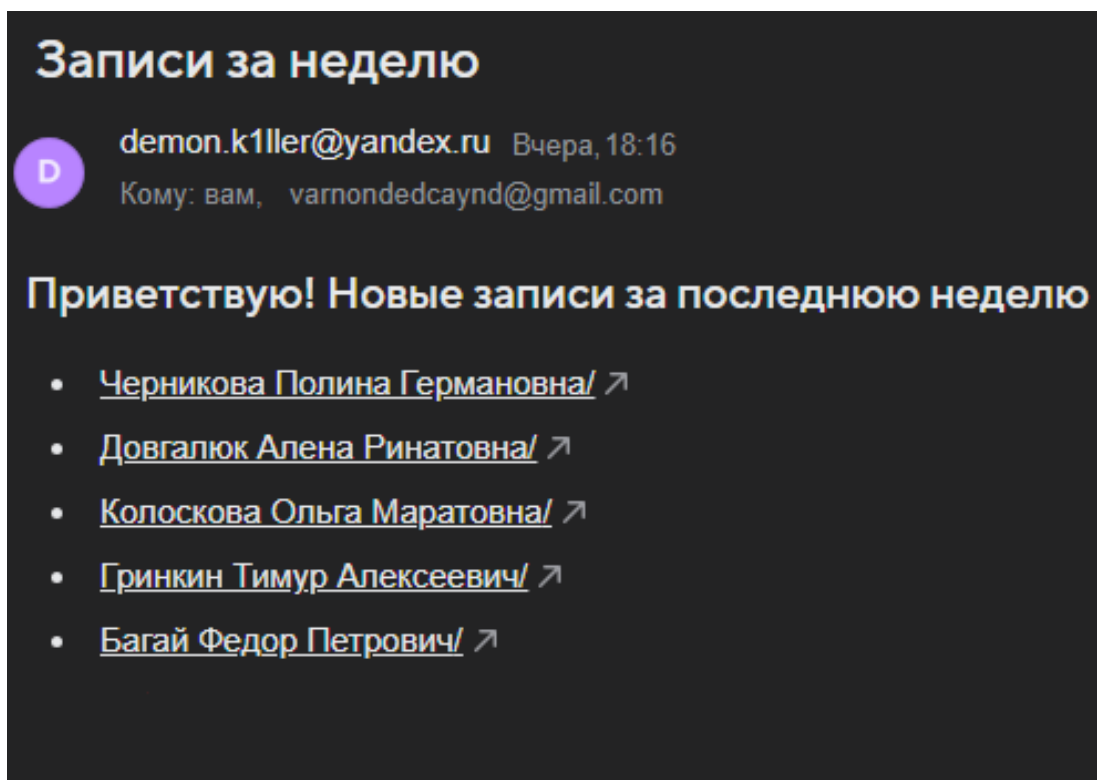


Рисунок 22 – Письмо с записями за неделю

4 ОПИСАНИЕ И ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

4.1 Врачи и распределение по кабинетам

При переходе на главную страницу пользователю с правами доступа представляется список всех работающих на данное время врачей и их принадлежность к определенному кабинету.

В данном приложении права доступа имеют все врачи, занесенные в базу данных, директор и администратор. В дальнейшем, пользователи с правами доступа будут именоваться, как пользователи.

Этот список можно редактировать и удалять в случае, если ФИО врача поменяется или человек уволился.

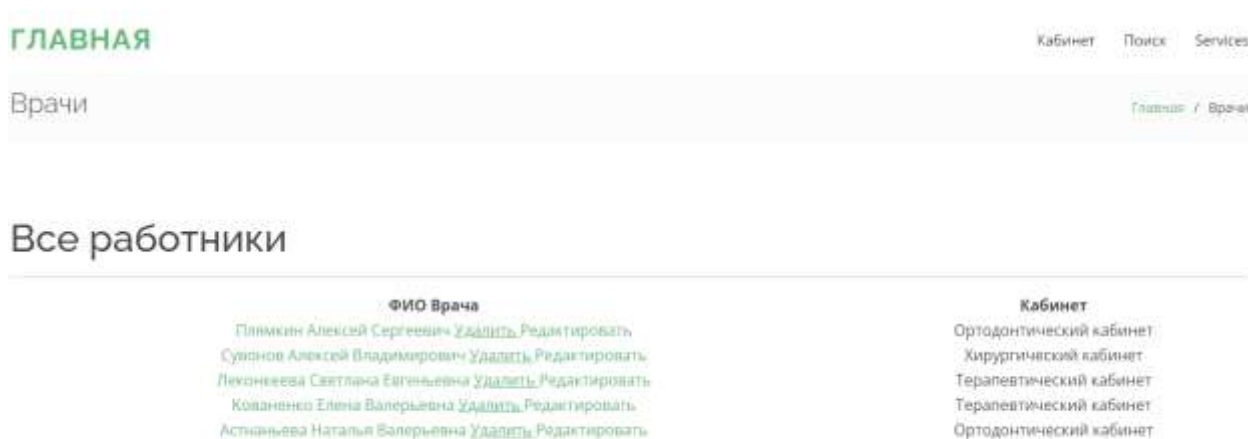


Рисунок 23 – Главная страница

4.2 Пациенты каждого врача

При нажатии на ФИО врача, выполняется переход на связанную с этим ФИО страницу, в ней отображена ранее показанная информация, а также список клиентов, проходивших лечение у данного врача.

Этот список так же можно редактировать или удалять, а также, на этой странице есть функция добавить запись о пациенте, которая будет занесена в эту таблицу.

Врач - Плянкин Алексей Сергеевич
Кабинет - Ортодонтический кабинет

• [Добавить запись](#)

ФИО Клиента	Оказанные Услуги	Итоговая сумма	Дата записи
Багай Федор Петрович Удалить Редактировать	Прием(осмотр,консультац...	3500.0	May 22, 2024, 12:17 a.m.
Гринин Тимур Алексеевич Удалить Редактировать	Лечение пульпита 1-канал...	4650.0	May 22, 2024, 12:16 a.m.
Довгалин Алена Ринатовна Удалить Редактировать	Лечение периодонтита 1-к...	4150.0	May 22, 2024, 12:14 a.m.
Колоскова Ольга Маратовна Удалить Редактировать	Лечение периодонтита пос...	12900.0	May 22, 2024, 12:15 a.m.
Черникова Полина Германовна Удалить Редактировать	Реминерализация зубов, ф...	10000.0	May 22, 2024, 12:12 a.m.

Рисунок 24 – Список пациентов

4.3 Создание, редактирование и удаление записей

При нажатии на кнопку «Добавить запись» открывается форма, в которой необходимо указать ФИО клиента, перечень оказанных услуг. При выборе услуг итоговая сумма будет складываться из цен, указанных ранее для каждой услуги. Дату записи так же указывать не нужно, она записывается автоматически, в моменте создания записи.

Имя клиента:

Услуги:

Услуги:

Услуги:

Услуги:

Стоимость:

Врач:

Подтверждение:

Рисунок 25 – Форма записи

При нажатии на кнопку «Редактировать» откроется та же форма записи, но уже заранее указанными при создании значениями. Необходимо только произвести необходимые изменения в записи и нажать на кнопку подтвердить. В таком случае значения в созданной до этого записи изменятся на те, что мы указали при изменении.

При нажатии на кнопку «Удалить» появится предупреждение и только после подтверждения запись удалиться.

Вы действительно хотите удалить данную запись?

подтвердить

Рисунок 26 – Предупреждение при удалении

4.4 Поиск записей по критериям

В шапке нашего приложения, справа-сверху, расположена кнопка «Поиск». При нажатии на нее, открывается список всех добавленных записей и форма поиска. Искать нужные записи можно по ФИО клиента, ФИО врача или даты, позже указанной.

После ввода критериев поиска приложение отобразит результаты, соответствующие запросу. Это делает процесс поиска более эффективным и удобным, позволяя быстро находить нужные данные и продолжать работу без лишних задержек.

Имя пациента:	<input type="text"/>	Врач:	Пляшкин	Даты позже:	дд.мм.гггг	<input type="checkbox"/>	Найти
ФИО Клиента	Оказанные Услуги	Итоговая сумма	Дата записи				
Багай Федор Петрович Удалить Редактировать	Прием(осмотр.консультац...	3500.0	May 22, 2024, 12:17 a.m.				
Гринкин Тимур Алексеевич Удалить Редактировать	Лечение пульпита 1-канал...	4650.0	May 22, 2024, 12:16 a.m.				
Довгалоюк Елена Ринатовна Удалить Редактировать	Лечение периодонтита 1-к...	4150.0	May 22, 2024, 12:14 a.m.				
Колоскова Ольга Маратовна Удалить Редактировать	Лечение периодонтита пос...	12900.0	May 22, 2024, 12:15 a.m.				
Прохоров Антон михайлович Удалить Редактировать	Прием(осмотр.консультац...	1500.0	June 12, 2024, 12:24 a.m.				
Черникова Полина Германовна Удалить Редактировать	Реминерализация зубов, Г...	10000.0	May 22, 2024, 12:12 a.m.				

Рисунок 27 – Поиск записей по критериям

5 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

5.1 Безопасность

Безопасность рабочего места означает отсутствие рисков для здоровья при продолжительной работе. Оно должно быть удобным и эргономичным. Технологический процесс включает длительную работу за компьютером сидя, что может повредить здоровье, если не соблюдать определенные нормы и правила, установленные государственными органами.

При организации рабочего места необходимо учитывать эргономику, которая изучает взаимодействие человека и других элементов системы. Соблюдение норм эргономики гарантирует безопасное рабочее место.

Особое внимание следует уделять основным предметам, которые при неправильном использовании могут нанести вред здоровью. К таким предметам относятся кресло, стол, пользовательский интерфейс и экран. Эргономика требует, чтобы мебель для работы сидя перед компьютером была удобной и доступной.

Проектирование удобного использования включает рассмотрение этого вопроса на всех этапах жизненного цикла продукта, от концепции до утилизации. Доступность означает, что продукция, системы и услуги могут быть использованы людьми с широким диапазоном возможностей для достижения целей в конкретных условиях.

Правильное использование кресла помогает предотвратить проблемы со здоровьем. Сиденье кресла должно быть на такой высоте, чтобы пользователь мог касаться ногами пола. Высота кресла зависит от высоты стола: при высоте стола 720-780 мм удобна высота кресла 420-480 мм. Спинка должна уменьшать давление на поясницу, а подлокотники должны быть настроены так, чтобы слегка касаться локтей, не давить и не поднимать их выше естественного положения. Если пользователь испытывает дискомфорт при правильной настройке мебели, это указывает на нарушение эргономики рабочего

места. Важно, чтобы рабочая поверхность не вызывала дискомфорта и не отвлекала от работы.

Стол должен обеспечивать достаточное рабочее пространство для размещения периферийных устройств и канцелярских предметов. Высота стола должна быть 750 мм, ширина и глубина должны соответствовать количеству оборудования на столе. Клавиатура должна находиться на столе. Если использование стола вызывает дискомфорт, необходимо принять меры, иначе возможны проблемы со здоровьем позвоночника.

Санитарные нормы устанавливают, что уровень шума на работе не должен превышать 70 децибел. Длительное воздействие шума выше этого уровня может вызвать проблемы со слухом и дезориентацию. Для решения этой проблемы при проектировании помещений используют звукоизоляционные материалы. Если ремонт невозможен, можно использовать стулья с высокими спинками, высокие шкафы и перегородки для уменьшения шума.

Монитор должен быть размещен так, чтобы верхний край экрана был на уровне глаз. Рекомендуемое расстояние от глаз до монитора составляет 50-70 см. Монитор следует разместить так, чтобы можно было время от времени фокусировать взгляд на удаленные объекты, предпочтительно у окна.

Для комфортного чтения информации с дисплея нужно соблюдать определенные значения яркости и контраста изображения, внешней освещенности экрана, размера знака и угла наблюдения. Яркость знаков должна быть не менее 20 кд/м² для плоских экранов, а контраст изображения – не менее 3:1.

Государственным стандартом Р 50949-2001 установлены методы измерения и оценки следующих эргономических параметров и параметров безопасности, регламентированных ГОСТ Р 50948-2001 [1]:

- яркость изображения;
- неравномерность яркости рабочего поля экрана;

- неравномерность яркости элементов знаков на плоских дискретных экранах;
- контрастность деталей изображения;
- контрастность изображения на плоских экранах в зависимости от угла наблюдения;
- контрастность соседних уровней кодирования яркостью;
- постоянство размера знака;
- ширина контура знака;
- пространственные искажения изображения по рабочему полю;
- пространственная нестабильность изображения (дрожания);
- несведение цветов;
- временная нестабильность изображения (мелькания);
- электростатический потенциал экрана;
- напряженность переменного электрического поля;
- плотность магнитного потока.

Следование правильным значениям каждого из параметров помогает снизить влияние плоских дискретных мониторов на здоровье пользователей.

Графический интерфейс пользователя должен быть разработан в соответствии с государственным стандартом, чтобы избежать проблем с эффективностью взаимодействия с системой. Все элементы интерфейса должны быть интуитивно понятными и не вызывать вопросов при их использовании. Если функцию элемента не удастся четко объяснить, следует привести пример его правильного использования.

В настоящее время интерфейс веб-сервиса включает следующие графические элементы:

- чек-бокс (рис. 28): используется для отметки положительного или отрицательного значения параметра;



Рисунок 28 – Чек-бокс подтверждения создания записи

– диалоговое окно (рис. 29): необходимо для оповещения пользователя о результате действия;

Вы действительно хотите удалить данную запись?

Рисунок 29 – Диалоговое окно о удалении записи

– поле ввода (рис. 30): необходимо для ввода данных при том или ином действии;

A rectangular input field with a black border. Inside the field, the text '123абв' is displayed in a blue font. A vertical cursor is positioned at the end of the text, after the letter 'в'.

Рисунок 30 – Поле ввода ФИО пациента

– поясняющая информация (рис. 31): необходимо для пояснения чего-либо в программе;

Страница авторизованного пользователя omega

На этой странице можно оставить заявку на прием к врачу

Two rectangular buttons with black borders. The first button contains the text 'Записаться' in blue. The second button contains the text 'Выйти' in blue.

Рисунок 31 – Поясняющая информация на домашней странице веб-сервиса

– кнопка списка/кнопка меню (рис. 32): используется для отображения элементов меню;

Three menu items are displayed. 'Кабинет' and 'Поиск' are in blue text. 'Главная' is in green text and is enclosed in a light gray rectangular box.

Рисунок 32 – Кнопка основного меню веб-сервиса

– нажимаемая кнопка (рис. 33): необходима для немедленного выполнения запроса;

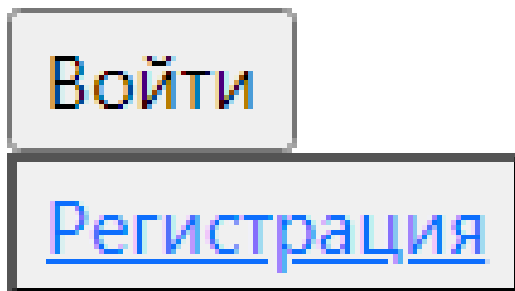


Рисунок 33 – Нажимаемая кнопка входа и регистрации

– таблица (рис. 34): используется для вывода большого числа информации на одну страницу;

Врач - Пляжкин Алексей Сергеевич
Кабинет - Ортодонтический кабинет

• [Добавить запись](#)

ФИО Клиента	Оказанные Услуги	Итоговая сумма	Дата записи
Багай Федор Петрович Удалить Редактировать	Прием/осмотр,консультац...	3500.0	May 22, 2024, 12:17 a.m.
Гранин Тимур Алексеевич Удалить Редактировать	Лечение пульпита 1-канал...	4650.0	May 22, 2024, 12:16 a.m.
Довгалюк Елена Ринатовна Удалить Редактировать	Лечение периодонтита 1-к...	4150.0	May 22, 2024, 12:14 a.m.
Колосова Ольга Маратовна Удалить Редактировать	Лечение периодонтита пос...	12900.0	May 22, 2024, 12:15 a.m.
Черникова Полина Германовна Удалить Редактировать	Реминерализация зубов, Г...	10000.0	May 22, 2024, 12:12 a.m.

Рисунок 34 – Таблица записей пациентов

Каждый элемент выполнен согласно ГОСТ Р ИСО 9241-161-2016, учитывая описание их действия, состояния и условия использования [3].

Графический интерфейс оформлен в светлых оттенках с выделением светло-зеленого цвета. Такая цветовая гамма позволяет пользователям легко воспринимать информацию на экране при любом освещении в помещении. Кроме того, светло-зеленый цвет вызывает у пользователей ассоциации с спокойствием, безопасностью и расслабленностью, что улучшает их взаимодействие с системой.

Помимо удобства использования, веб-сервис также защищен системой информационной безопасности. Доступ к базе данных предоставляется толь-

ко пользователям с необходимым уровнем прав, а данные зашифрованы для защиты от несанкционированного доступа. Например, данные учетной записи при регистрации преобразуются в хэш и хранятся в таком виде в базе данных.

5.2 Экологичность

Для эффективного использования веб-сервиса необходим стабильный и быстрый доступ к Интернету. При каждом запуске веб-сервиса происходит полная передача данных с сервера на устройство пользователя, что существенно увеличивает потребление интернет-трафика. Чем тяжелее страницы сайта, тем больше трафика требуется для их загрузки, что приводит к значительным выбросам CO₂ в атмосферу.

Чтобы уменьшить выбросы CO₂, рекомендуется применять масштабируемую архитектуру, выбирать экологичных провайдеров, использовать подходящие языки программирования и оптимизировать веб-страницы. Масштабируемая архитектура позволяет отключать неактивные серверы при низкой нагрузке, что снижает энергопотребление и повышает устойчивость сайта к сбоям.

Интернет-трафик часто проходит через отдаленные точки мира, что вызывает задержки и увеличивает энергопотребление серверов. Чтобы избежать этого, следует выбирать экологичных провайдеров или интернет-поставщиков, которые предоставляют возможность выбирать серверы, работающие на возобновляемых источниках энергии.

Выбор правильных языков программирования может не только сократить объем передаваемых данных, но и улучшить производительность веб-сайта. В настоящее время средний размер веб-страницы превышает 3 мегабайта, что значительно больше по сравнению с началом 2000-х годов, и этот показатель продолжает расти. Для уменьшения размеров страниц рекомендуется минимизировать количество неиспользуемого кода и оптимизировать методы программирования.

Изображения на сайтах являются основным источником углеродного следа, составляя в среднем 60% всего объема ресурсов. Их оптимизация может существенно снизить нагрузку на сайт. Многие изображения сохраняются в неподходящих форматах, имеют некорректные размеры или плохо оптимизированы.

На этапе разработки, когда определяется пользовательское взаимодействие, стоит устранить элементы, требовательные к ресурсам. Следует избегать слишком сложных интерфейсных элементов, таких как карусели, которые содержат много данных и требуют JavaScript, что увеличивает общий вес страницы.

Также необходимо пересмотреть необходимость размещения на сайте кнопок для публикации в социальные сети, встраиваемых карт, видео с автоматическим воспроизведением, Flash-контента, рекламных баннеров и встроенного контента от третьих лиц, так как все они требуют загрузки большого объема дополнительных данных.

Чтобы сделать Интернет более доступным и снизить его воздействие на окружающую среду рекомендуется:

- выбирать хостинг, пользующийся серверами, работающими на возобновляемых источниках энергии;
- разрабатывать веб-сайт с нуля, а не с шаблона;
- отдельно сохранять и архивировать неактуальный контент;
- оптимизировать большие файлы (картинки и аудио).

5.3 Чрезвычайные ситуации

В процессе использования веб-сервиса могут возникнуть различные чрезвычайные ситуации, такие как:

- потеря данных;
- проблемы с безопасностью;
- проблемы с производительностью.

Чтобы предотвратить потерю данных пользователей, необходимо регулярно создавать резервные копии базы данных. Это позволит быстро восстановить информацию в случае её утраты. Чрезвычайные ситуации не возникают без причины; они являются следствием плохого контроля состояния базы данных. Важно регулярно проводить диагностику для выявления и устранения неисправностей.

Наиболее распространённые уязвимости веб-сайтов включают:

- некорректная настройка параметров безопасности;
- межсайтовое выполнение сценариев (XSS);
- недостатки аутентификации;
- недостатки контроля доступа;
- внедрение вредоносного кода.

Помимо проблем с базой данных, веб-сервисы могут быть подвергнуты атакам злоумышленников, что может привести к утечке конфиденциальной информации. Для предотвращения таких атак необходимо продумать систему безопасности и иметь план действий на случай их возникновения.

Для защиты от межсайтового выполнения сценариев (XSS) следует экранировать пользовательский ввод, заменяя специальные символы безопасными. Высокий уровень безопасности веб-приложения достигается за счёт следующих мер:

- быстрое исправление выявленных уязвимостей;
- автоматизация процессов, где это возможно.

Проблемы с производительностью могут проявляться в виде:

- некорректной загрузки веб-страниц;
- низкой скорости работы элементов интерфейса;
- долгой обработки запросов.

Для предотвращения таких проблем рекомендуется:

- загружать контент по мере необходимости, чтобы не нагружать пользователя лишней информацией;

– использовать отдельные файлы для языков программирования, чтобы загружать их содержимое один раз и потом использовать по мере необходимости;

– применять системы кэширования для уменьшения нагрузки на сервер;

– не использовать изображения для отображения текста;

– избегать изменения размеров изображений в HTML, используя их исходные размеры;

– оптимизировать код, применяя универсальный тег `<div>` и стили CSS вместо множества HTML-тегов.

Следование этим рекомендациям поможет поддерживать высокий уровень безопасности и производительности веб-сервиса, минимизируя вероятность возникновения проблем

ЗАКЛЮЧЕНИЕ

В ходе практики было разработано финансовое веб-приложение для стоматологической клиники ООО "Зубная архитектура". Основной целью этого приложения стало обеспечение эффективного управления финансовой деятельностью клиники и повышение удобства для персонала.

При разработке приложения был уделен особый акцент на соблюдение современных требований безопасности и функциональности. Кроме того, веб-приложение было разработано с учетом потребностей персонала клиники, обеспечивая удобный и интуитивно понятный интерфейс.

Это позволяет сотрудникам клиники легко управлять финансовыми процессами, вносить данные и получать необходимую финансовую информацию, что способствует повышению производительности и эффективности работы клиники.

Таким образом, разработанное финансовое веб-приложение стало важным инструментом для управления финансами и повышения удобства работы персонала стоматологической клиники ООО "Зубная архитектура".

В рамках практики были успешно реализованы следующие ключевые функции:

- возможность авторизации и регистрации пользователей: пользователи имеют возможность безопасно авторизоваться в системе и зарегистрироваться, что обеспечивает защиту доступа к данным;
- добавление, редактирование и удаление записей: приложение позволяет оперативно вносить изменения в базу данных клиники, что существенно упрощает управление информацией;
- поиск по различным критериям: пользователи могут осуществлять поиск по различным параметрам, таким как дата, имя врача, имя пациента, что облегчает навигацию и поиск необходимой информации;

- добавление возможности клиентам оставлять заявки на запись: это облегчить запись для клиентов, а также упростить работу с клиентами для персонала;

- рассылка отчетов о занесенных записях на указанные почтовые адреса: приложение позволяет автоматически рассылать отчеты о занесенных записях на указанные почтовые адреса, обеспечивая внутреннюю отчетность в клинике;

- кеширование данных: для оптимизации производительности приложения и снижения нагрузки на сервер используется кеширование данных;

- система отслеживания ошибок: реализована система отслеживания ошибок, которая позволяет оперативно выявлять и устранять проблемы в работе приложения.

Таким образом, разработанное веб-приложение представляет собой мощный инструмент для управления финансовой деятельностью стоматологической клиники. Оно обеспечивает эффективное ведение учета и обработку данных, что способствует повышению производительности и улучшению качества работы клиники.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Воронина, Л. П. Управление финансами учреждения здравоохранения / Л.П. Воронина // Муниципалитет: экономика и управление. – 2019. – №3 (28). – С 57– 67
- 2 ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению. 1978.
- 3 ГОСТ 24.701-86. Единая система стандартов автоматизированных систем управления. Надежность автоматизированных систем управления. Основные положения. М.: Издательство стандартов, 1987. – 17 с.
- 4 ГОСТ Р 50949-2001. Средства отображения информации индивидуального пользования. Методы измерений и оценки эргономических параметров и параметров безопасности. М., 2002. 27 с.
- 5 ГОСТ Р 50948-2001. Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности. М., 2002. 13 с.
- 6 ГОСТ Р ИСО 9241-161-2016. Эргономика взаимодействия человек-система. Часть 161. Элементы графического пользовательского интерфейса. М., 2016. 58 с.
- 7 ГОСТ Р 56274-2014. Общие показатели и требования в эргономике. М., 2015. 31 с.
- 8 Негуч, Н. А. Обзор и сравнение программных продуктов для анализа финансового состояния предприятия / Н. А. Негуч. – Текст : непосредственный // Молодой ученый. – 2019. – № 21 (259). – С. 229 – 233.
- 9 Технологии бережливого производства в совершенствовании амбулаторной медицинской помощи: учебное пособие / В.А. Бойков, С.В. Барановская, И.А. Деев и [др.]. – Томск: Изд-во СибГМУ, 2021. – 94 с.

10 Сингхал, Ш. Тренды в цифровизации здравоохранения / Ш. Сингхал, А. Маклеод, Н. Бансал [Текст] // McKinsey в здравоохранении. — :McKinsey&company, 2023. — С. 8.

11 Garcia, Manuel Исползованию информационных технологий в стоматологии / Garcia, Manuel [Электронный ресурс] // Stanford University : [сайт]. — Режим доступа : https://www.stanford.edu/search/?q=Use+of+information+technology+in+dentistry&search_type=web&submit=. — 10.05.2024.

ПРИЛОЖЕНИЕ А

Техническое задание

1 Введение

1.1 Наименование программы

Наименование программы – «Финсчет».

1.2 Краткая характеристика области применения

Система «Финсчет» предназначена для подсчета прибыли за день и хранения её в базе данных. Предусмотрено, что бухгалтер будет каждый день заносить данные в программу, тем самым пополняя базу данных.

2 Основания для разработки

Основанием для разработки является устный договор с директором ООО «Зубная Архитектура», именуемым в дальнейшем Заказчиком и Бедновой Дарьей, именуемой в дальнейшем исполнителем.

Согласно Договору, Исполнитель обязан разработать и установить систему «Финсчет» на оборудовании Заказчика, предоставить исходные коды и документацию к разработанной системе.

Наименование темы разработки – «Разработка финансового приложения для стоматологической клиники».

3 Назначение разработки

Программа будет использоваться в стоматологической клинике двумя людьми: бухгалтером и директором.

3.1 Функциональное назначение

Для бухгалтера программа предоставляет возможность просмотра, записи, редактирования, удаления данных о зарплате в текущий день и просмотра записей за предыдущие дни.

Для директора программа предоставляет возможность просмотра общей прибыли и поделенной на дни\недели\месяцы\года.

Продолжение ПРИЛОЖЕНИЯ А

3.2 Эксплуатационное назначение

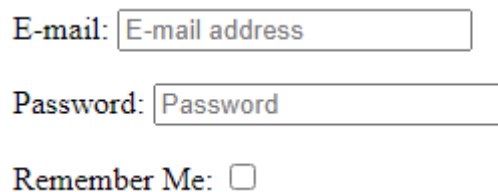
Программа должна эксплуатироваться в бухгалтерском отделе. С правами бухгалтера программа запускается на компьютере бухгалтера.

4 Требования к программе или программному изделию

4.1 Требования к функциональным характеристикам

4.1.1 Требования к составу выполняемых функций

После запуска программы пользователю отображается форма ввода логина и пароля.



The image shows a login form with three elements: an 'E-mail:' label followed by a text input field containing the placeholder 'E-mail address'; a 'Password:' label followed by a text input field containing the placeholder 'Password'; and a 'Remember Me:' label followed by an unchecked checkbox.

Рисунок 35 – Ввод логина и пароля.

В системе существует всего 2 пользователя — бухгалтер и директор. Программа проверяет тип пользователя и открывает соответствующий интерфейс.

Для бухгалтера программа предоставляет следующие возможности:

- запись на текущий день прибыли;
- редактирование записи на текущий день;
- удаление записи на текущий день;
- просмотр всей прибыли за все записанные дни;
- при просмотре выводится таблица, каждая строка которой содержит

следующую информацию:

- 1) дата и время записи;
- 2) название процедуры;
- 3) стоимость процедуры.

Продолжение ПРИЛОЖЕНИЯ А

Так как таблицы могут выходить громоздкими, будет функция поиска, по дате и названию процедуры.

Для директора будет доступ не только к записям, но и к доступу к админ-панели, где он может назначать права для бухгалтера, просматривать сеансы. Пример админ панели можно увидеть на рисунке 2.



Рисунок 36 – Админ-панель

Окно таблицы финансов должны содержать кнопки поиск, редактировать, добавить и удалить.

Для удаления нужно будет нажать кнопку «подтвердить» для предотвращения удалений по неосторожности.

Для изменения записей, нужно нажать кнопку «редактировать» и после внесения изменений нажать подтверждение.

Для добавления записи бухгалтер нажимает кнопку «Добавить», вносит всю нужную информацию и добавляет запись. Если введены некорректные

Продолжение ПРИЛОЖЕНИЯ А

значения — то запись можно отредактировать. Введения букв в поле для суммы процедуры выведет ошибку.

Все изменения, выполняемые бухгалтером, должны отражаться на админ-панели у директора.

4.1.2 Требования к организации входных и выходных данных

Данные о процедурах и их стоимость хранятся в базе данных. Ввод данных в базу (логины и пароли) выполняет директор, по инструкции предоставляемой исполнителем. Выполняет он это с помощью запросов к СУБД. После установки программы, ввод данных в систему осуществляет только бухгалтер:

- дата и время должны быть записаны в формате: «ДД.ММ.ГГГГ ЧЧ:ММ»;
- название — последовательность не более чем из 200 любых символов;
- сумма – поле с ограничением на ввод только цифр.

4.1.3 Требования к временным характеристикам

После изменения бухгалтером данных, находящихся в базе данных, новая информация отображается не позднее, чем через 5 секунд.

4.2 Требования к надежности

Вероятность безотказной работы системы должна составлять не менее 99.99% при условии исправности сети.

4.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы

В связи с тем, что в базе данных хранятся данные о выручке за день(финансовая информация) — базу данных стоит резервировать (резервирование замещением).

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

Продолжение ПРИЛОЖЕНИЯ А

- организацией бесперебойного питания технических средств;
- использованием лицензионного программного обеспечения;
- регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;
- регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов.

4.2.2 Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 10 минут при условии соблюдения условий эксплуатации технических и программных средств. Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

4.2.3 Отказы из-за некорректных действий оператора

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу пользователя без предоставления ему административных привилегий.

4.3 Условия эксплуатации

Программа (клиент) запускается на компьютере бухгалтера и компьютере директора. База данных находится на третьем компьютере. Должна

Продолжение ПРИЛОЖЕНИЯ А

существовать устойчивая связь по сети между клиентами и базой данных. Окно программы может быть свернуто или закрыто на первых двух компьютерах, на третьем компьютере с базой данных программа должна быть всегда запущена. Программа может работать на фоне других приложений, не нарушая своей работы.

4.3.1 Климатические условия эксплуатации

Специальные условия не требуются.

4.3.2 Требования к видам обслуживания

Программа не требует проведения каких-либо видов обслуживания.

4.3.3 Требования к численности и квалификации персонала

При установке и настройке системы необходим системный администратор. В процессе эксплуатации с программой работают директор и бухгалтер.

Системный администратор должен иметь необходимые знания для установки и настройки приложения. В перечень задач, выполняемых системным администратором, должны входить:

- установка клиентских приложений;
- настройка СУБД;
- настройка сети между клиентами и СУБД.

Пользователь программы (оператор) должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

Директор и бухгалтер должны быть прошли инструктаж по электробезопасности (для работы с конторским оборудованием). Проводить инструктаж обязан администратор.

4.4 Требования к составу и параметрам технических средств

Состав технических средств:

- Компьютер бухгалтера, включающий в себя:

Продолжение ПРИЛОЖЕНИЯ А

1) процессор x86 с тактовой частотой, не менее 1 ГГц;

2) оперативную память объемом, не менее 1 Гб;

3) видеокарту, монитор, мышь, клавиатура.

– Компьютер директора, включающий в себя:

1) процессор x86 с тактовой частотой, не менее 1 ГГц;

2) оперативную память объемом, не менее 1 Гб;

3) видеокарту, монитор, мышь, клавиатура.

– Два компьютера для СУБД (основной и резервный), включающий в себя:

1) процессор x86 с тактовой частотой, не менее 1 ГГц;

2) оперативную память объемом, не менее 1 Гб;

3) видеокарту, монитор, мышь, клавиатура.

4.5 Требования к информационной и программной совместимости

Приложения бухгалтера и директора обмениваются с СУБД сообщениями по локальной сети, при этом используется протокол HTTP. Должно быть исключено появление посторонних устройств в сети.

4.6 Требование к маркировке и упаковке

Программное изделие передается по внешнему носителю. Специальных требований к маркировке не предъявляется. Для проверки подлинности программного обеспечения рекомендуется проверять контрольные суммы загруженных файлов со значениями, подсчитанными перед эксплуатацией.

4.7 Требования к транспортированию и хранению

Специальных требований не предъявляется.

4.8 Специальные требования

Программа должна обеспечивать взаимодействие с пользователем посредством графического пользовательского интерфейса, разработанного согласно пожеланиям заказчика.

Продолжение ПРИЛОЖЕНИЯ А

5 Требования к программной документации

Предварительный состав программной документации:

- техническое задание (включает описание применения);
- программа и методика испытаний;
- руководство системного программиста;
- руководство оператора;
- руководство программиста;
- ведомость эксплуатационных документов;
- формуляр.

6 Техничко-экономические показатели

Программа «Финсчет» предназначена для небольших стоматологических клиник, в которых бухгалтерский отдел не больше одного-двух человек. Скорее всего программа будет использоваться как программа отдельного учета прибыли врачей. Функциональность программы совпадает с аналогами.

В связи с тем, что существуют программы стоматологических клиник с учетом, который частично выполняют функции подсчета финансов, ждать масштабного распространения не стоит. Однако, в случае бесплатного распространения программы, потребность в ней может быть весьма высокой из-за восполнения функций учета, не доступных для программ для стоматологических клиник. Экономический эффект при этом может быть обеспечен за счет платной установки системы.

7 Стадии и этапы разработки

Разработка должна быть проведена в три стадии:

- техническое задание;
- технический (и рабочий) проекты;
- внедрение.

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

Продолжение ПРИЛОЖЕНИЯ А

На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:

- разработка программы;
- разработка программной документации;
- испытания программы.

На стадии «Внедрение» должен быть выполнен этап разработки «Подготовка и передача программы».

Содержание работ по этапам:

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

- постановка задачи;
- определение и уточнение требований к техническим средствам;
- определение требований к программе;
- определение стадий, этапов и сроков разработки программы и документации на нее;
- согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

- разработка, согласование и утверждение порядка и методики испытаний;
- проведение приемо-сдаточных испытаний;
- корректировка программы и программной документации по результатам испытаний.

Продолжение ПРИЛОЖЕНИЯ А

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах заказчика.

8 Порядок контроля и приемки

Приемосдаточные испытания программы должны проводиться согласно разработанной исполнителем и согласованной заказчиком «Программы и методики испытаний».

Ход проведения приемо-сдаточных испытаний заказчик и исполнитель документируют в протоколе испытаний.

На основании протокола испытаний исполнитель совместно с заказчиком подписывают акт приемки-сдачи программы в эксплуатацию.

Список используемой литературы

1. ГОСТ 19.201-78 Единая система программной документации. Техническое задание. Требования к содержанию и оформлению. 1978. Режим доступа: <http://protect.gost.ru/document.aspx?control=7&id=155153>
2. ГОСТ 24.701-86. Единая система стандартов автоматизированных систем управления. Надежность автоматизированных систем управления. Основные положения. М.: Издательство стандартов, 1987. — 17 с.