

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика

(код и наименование направлению подготовки / специальности)

Разработка программного обеспечения

(направленность (профиль) / специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему Разработка программного обеспечения веб-представительства страхового агентства

Обучающийся

В.С. Околин

(Инициалы Фамилия)

(личная подпись)

Руководитель

д.т.н, доцент С. В. Мкртычев

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

к.п.н, доцент С.А. Гудкова

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2025

Аннотация

Тема: Разработка Веб-представительства страхового агентства

Ключевые слова: веб-разработка, Flask, Python, страховое агентство.

Выпускная квалификационная работа посвящена разработке программного обеспечения для веб-представительства страхового агентства. Целью проекта являлось создание удобного и функционального веб-сайта, позволяющего клиентам просматривать каталог страховых продуктов, рассчитывать стоимость полиса онлайн, оформлять заявки на страхование и управлять заказами через личный кабинет. Для администраторов реализованы инструменты управления продуктами, офисами и заявками.

В ходе работы проведен анализ предметной области, изучены существующие решения, сформулированы требования к системе, выполнено проектирование с использованием UML-диаграмм. Реализация включает фронтенд на HTML/CSS/JavaScript и бэкенд на Python с использованием фреймворка Flask и СУБД SQLite.

Работа состоит из трёх глав, заключения, списка литературы и приложений:

В первой главе проводится анализ предметной области, определяются требования к системе и рассматриваются существующие решения.

Во второй главе описывается проектирование системы: диаграммы, модель данных и структура приложения.

В третьей главе представлены реализация веб-представительства, тестирование.

В заключении подводятся итоги и предлагаются пути дальнейшего развития системы.

Объем работы – 65 страниц, включая 3 таблицы, 30 рисунков и приложение. Список используемой литературы и используемых источников содержит 21 источник.

Abstract

The title of the graduation work is: Development of a Web Presence for an Insurance Agency.

The graduation work consists of three chapters, a conclusion, a list of references including 21 sources, and appendices with 30 figures and 3 tables. The aim of this graduation work is to develop a functional website for an insurance agency that allows customers to browse products, calculate policy costs, submit applications, and manage orders via a personal account. The system also provides administrators with tools for managing products, offices, and applications. The object of the graduation work is the digital representation of an insurance agency through a web platform.

The subject of the graduation work is the design and implementation of a web application using HTML/CSS/JavaScript for the frontend and Python with the Flask framework and SQLite database for the backend.

The key issue of the work is to create a reliable and user-friendly solution for customer interaction and internal management in the insurance business. The work includes analysis of existing solutions, system design with UML diagrams, and implementation of the web application with testing procedures. The first part describes the current state of web technologies in the insurance industry and defines functional requirements. The second part outlines the system architecture, data models, and interface design.

The third part presents the development and testing of the web application. In conclusion, the developed system meets the defined requirements and provides effective tools for managing insurance services.

Further improvements include expanding functionality, enhancing security, and using more scalable technologies.

The work is of interest for a wide circle of readers interested in web development for the insurance industry using modern technologies such as Flask and Python.

Оглавление

Введение	5
Глава 1 Постановка задачи на разработку программного обеспечения веб-представительства страхового агентства	7
1.1 Техничко-экономическая характеристика предметной области.....	7
1.2 Разработка требований к веб-представительству страхового агентства.....	8
1.3 Анализ существующих веб-представительств страховых компаний ..	10
Глава 2 Проектирование компонентов веб-представительства страхового агентства	18
2.1 Разработка логической модели веб-представительства	18
2.2 Логическая модель данных веб-представительства	24
2.3 Разработка структуры веб-представительства страхового агентства	27
Глава 3 Реализация веб-представительства страхового агентства	32
3.1 Разработка графического дизайна веб-представительства	32
3.2. Обоснование архитектуры веб-представительства	34
3.3 Средства реализации веб-представительства.....	38
3.4 Физическая модель данных веб-представительства	40
3.5 Маршрутизация веб-страниц приложения	42
3.6 Проектирование и разработка пользовательского интерфейса	44
3.7 Размещение веб-представительства на хостинге.....	47
3.8 Тестирование Jmeter	49
3.9 Тестирование ZAP 2.16.1.....	51
Заключение.....	55
Список используемой литературы и используемых источников.....	58
Приложение А Изображения интерфейса веб-представительства.....	61

Введение

Страхование играет важную роль в защите имущественных интересов граждан и организаций. Оно охватывает такие направления, как автострахование (ОСАГО, КАСКО), страхование жизни, здоровья, имущества и ответственности. Для успешной работы страховым агентствам необходим удобный инструмент взаимодействия с клиентами – например, веб-представительство [8].

С развитием цифровых технологий всё больше людей предпочитают получать услуги онлайн. Это делает создание современного веб-представительства актуальным для страховых компаний. Такое программное обеспечение (ПО) позволяет автоматизировать процессы, улучшить обслуживание клиентов и выделиться среди конкурентов.

Актуальность темы разработки веб-представительства страхового агентства обусловлена следующими причинами:

- автоматизация процессов веб-представительства помогает снизить затраты времени и усилий на оформление заявок и расчёт стоимости полисов;
- конкуренция – современное веб-представительство становится важным фактором привлечения и удержания клиентов;
- рост числа интернет-пользователей – более 70% населения развитых стран ищут товары и услуги в интернете, поэтому страховые компании должны адаптироваться к этому;
- технологии – использование современных инструментов, таких как Flask, HTML/CSS/JavaScript и баз данных, позволяет создавать надёжные и масштабируемые решения [2][9].

Объектом исследования является веб-представительство страхового агентства.

Предметом исследования является программное обеспечение веб-представительства страхового агентства.

Целью работы является разработка программного обеспечения веб-представительства страхового агентства.

Задачи исследования:

- сформулировать требования и выполнить постановку задачи на разработку ПО веб-представительства страхового агентства;
- выполнить проектирование ПО веб-представительства страхового агентства;
- реализовать ПО веб-представительства страхового агентства и провести его тестирование.

Методы исследования: методы и технологии разработки веб-приложений.

Работа состоит из трёх глав, заключения, списка литературы и приложений:

В первой главе проводится анализ предметной области, определяются требования к системе и рассматриваются существующие решения.

Во второй главе описывается проектирование системы: диаграммы, модель данных и структура приложения.

В третьей главе представлены реализация веб-представительства, тестирование.

В заключении подводятся итоги и предлагаются пути дальнейшего развития системы.

Объем работы – 65 страниц, включая 3 таблицы, 30 рисунков и приложение.

Список используемой литературы и используемых источников содержит 21 источник.

Глава 1 Постановка задачи на разработку программного обеспечения веб-представительства страхового агентства

1.1 Технико-экономическая характеристика предметной области

Страхование является одним из ключевых секторов финансового рынка, который обеспечивает защиту имущественных интересов граждан и организаций. Основная цель страхования – минимизация рисков, связанных с непредвиденными событиями (например, аварии, болезни, стихийные бедствия). Современные страховые компании предлагают широкий спектр услуг, включая:

- автострахование (ОСАГО, КАСКО);
- страхование жизни и здоровья;
- имущественное страхование (недвижимость, техника);
- страхование ответственности (гражданская, профессиональная).

Развитие цифровых технологий привело к трансформации страховой отрасли. Сегодня страховые компании активно внедряют онлайн-сервисы для упрощения взаимодействия с клиентами, автоматизации процессов оформления полисов и расчета выплат.

Внедрение информационных систем в страховую отрасль позволяет достичь значительных экономических выгод:

- снижение операционных затрат – уменьшение количества сотрудников, занятых рутинными задачами (например, расчет стоимости полисов), сокращение времени на обработку заявок за счет автоматизации;
- увеличение доходов – расширение клиентской базы за счет онлайн-продаж, повышение лояльности клиентов благодаря удобному пользовательскому интерфейсу;
- минимизация рисков – более точный расчет страховых тарифов на основе анализа больших данных, снижение вероятности

мошенничества за счет использования современных методов проверки данных.

Таким образом, цифровая трансформация играет важную роль в развитии страховой отрасли, обеспечивая ее устойчивый рост, повышение конкурентоспособности и улучшение качества предоставляемых услуг [8][12][15].

1.2 Разработка требований к веб-представительству страхового агентства

«Для более точного определения требований к веб-представительству страхового агентства – таким образом, чтобы они соответствовали бизнес-целям компании и при этом учитывали потребности пользователей» [13] – может быть применена методология FURPS+. Эта подход позволяет классифицировать требования по пяти ключевым категориям:

- Functionality (Функциональные требования) – что должна делать система;
- Usability (Требования к удобству использования) – определяет, насколько система удобна и интуитивно понятна для пользователей;
- Reliability (Надежность) – способность системы обеспечивать стабильную и предсказуемую работу;
- Performance (Производительность) – как быстро и эффективно система выполняет свои функции;
- Supportability (Поддержка) – возможность поддержки, обновления и адаптации системы.

Ниже приведена таблица 1 требований к ПО веб-представительства страхового агентства с точки зрения методологии FURPS+.

Таблица 1 – Требования к веб-представительству в методологии FURPS+

Требование	Статус	Полезность	Риск	Стабильность
Функциональные требования				
Главная страница с краткой информацией о компании	Обязательное	Важная	Низкий	Высокая
Калькулятор расчета стоимости страхового полиса	Обязательное	Критическая	Средний	Высокая
Возможность онлайн-оформления (подбор полиса)	Обязательное	Критическая	Высокий	Средняя
Личный кабинет для клиентов (полисы, платежи)	Обязательное	Критическая	Высокий	Средняя
Требования к удобству использования				
Адаптивный дизайн для всех типов устройств	Обязательное	Критическая	Средний	Средняя
Простая навигация и понятный интерфейс	Обязательное	Важная	Средний	Средняя
Надежность				
Доступность 24/7	Обязательное	Критическая	Средний	Высокая
Время простоя (сбоя) в день – не более 5 минут	Обязательное	Важная	Средний	Средняя
Производительность				
Время отклика сервера – до 2 секунд	Обязательное	Важная	Средний	Средняя
Скорость генерации документов (полисов) – до 5 с	Обязательное	Важная	Средний	Средняя
Поддержка				
Удобство расширения функционала (новые виды полисов)	Обязательное	Важная	Средний	Средняя
Время устранения критических ошибок – до 1 часа	Обязательное	Критическая	Средний	Средняя

«Таблица требований к программному обеспечению предоставляет четкое представление о том, какие аспекты должны быть учтены при разработке системы. Каждая категория требований (функциональные, удобство, надежность, производительность, поддержка) играет важную роль в создании качественного продукта. Правильный баланс между всеми категориями требований гарантирует

создание программного продукта, который будет эффективным, удобным и надежным для пользователей» [14].

1.3 Анализ существующих веб-представительств страховых компаний

Для разработки качественного веб-представительства страхового агентства важно провести анализ существующих решений на рынке. Это позволит выявить лучшие практики, определить сильные и слабые стороны конкурентов, а также сформировать требования к будущему проекту, реализация которых даст веб-представительству конкурентное преимущество [4] [17].

На рынке представлено множество страховых компаний, каждая из которых имеет собственное веб-представительство. Рассмотрим некоторые из них:

Рассмотрим веб-представительство компании «Росгосстрах» демонстрирует высокий уровень функциональности, удобства и безопасности. Пользователям доступен интуитивно понятный калькулятор страхования, позволяющий рассчитать стоимость полисов для различных продуктов, таких как ОСАГО, КАСКО и страхование жизни. Веб-представительство также поддерживает полноценное онлайн-оформление полисов без необходимости посещения офиса (рисунок 1).

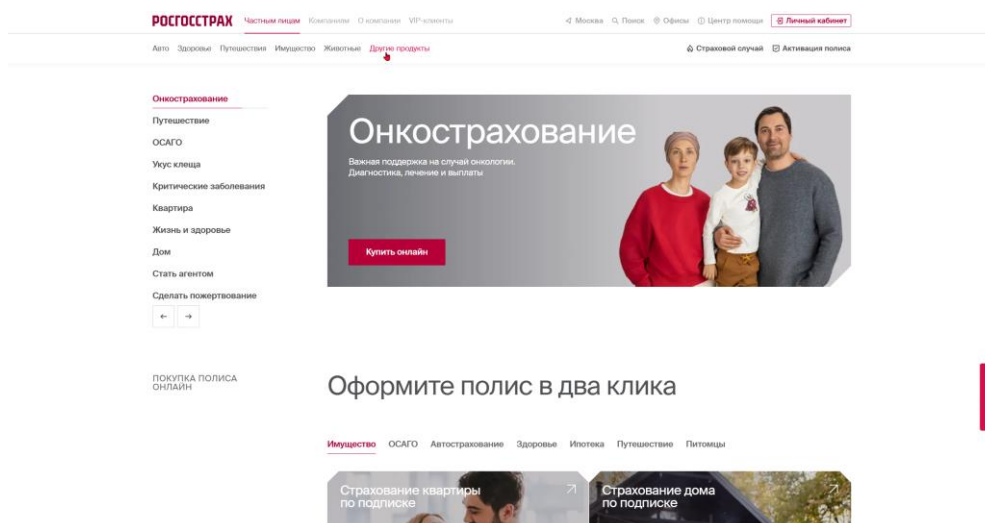


Рисунок 2 – Главная страница сайта

Производительность заслуживает высокой оценки. Платформа способна выдерживать до 500 одновременных запросов без потери стабильности, что особенно важно в периоды высокой нагрузки. Благодаря этим характеристикам, сайт «Росгосстрах» является образцовым примером современного страхового портала.

Веб-представительство страховой компании «Ингосстрах» предлагает базовый функционал для расчета стоимости страховых продуктов и оформления заявок. На сайте представлен калькулятор, охватывающий основные виды страхования, однако его интерфейс менее интуитивен по сравнению с конкурентами (рисунок 3).

Страхование квартиры

Расчет 4 430 ₽

0 ₽ — 5 000 000 ₽

Ответственность перед соседями 100 000 ₽

0 ₽ — 5 000 000 ₽

Стены и перекрытия 1 000 000 ₽

0 ₽ — 25 000 000 ₽

Движимое имущество 50 000 ₽

0 ₽ — 3 000 000 ₽

Дополнительная защита

Военные риски 775 ₽

Воздействие атмосферных осадков 1 100 ₽

Ответственность перед соседями при ремонте 510 ₽

Личные вещи 30 000 ₽ 800 ₽

Промокод Применить Продолжить

Экономия до 90% При оплате полисов Ингосстрах Авторизоваться

Параметры полиса

Отделка и инженерное оборудование	500 000 ₽
Ответственность перед соседями	100 000 ₽
Стены и перекрытия	1 000 000 ₽
Движимое имущество	50 000 ₽

Какие риски покрывает полис

Оплата в месяц Оплата в год

Выгода при покупке годового полиса 10%

Детали

Оплачиваемый полис	3 630
Личные вещи	800

Итого 4 430 ₽

Рисунок 3 – Калькулятор расчета стоимости страхования квартиры

Оформление заказов доступно только частично – после заполнения формы требуется дополнительная регистрация и выполнение ряда шагов, что усложняет процесс для пользователей. Личный кабинет предоставляет минимальные возможности для управления полисами, просмотра истории, и его использование затруднено из-за перегруженного и неструктурированного интерфейса.

С точки зрения удобства использования сайт имеет ряд недостатков. Навигация сложна и содержит лишние элементы, что мешает быстрому поиску информации (рисунок 4). Адаптивный дизайн реализован лишь частично – на мобильных устройствах интерфейс разрушается. Скорость загрузки страниц также невысока, особенно при высоких нагрузках. Тестирование показало, что сайт способен выдерживать до 300 одновременных запросов, но при превышении этой отметки возникают ошибки.

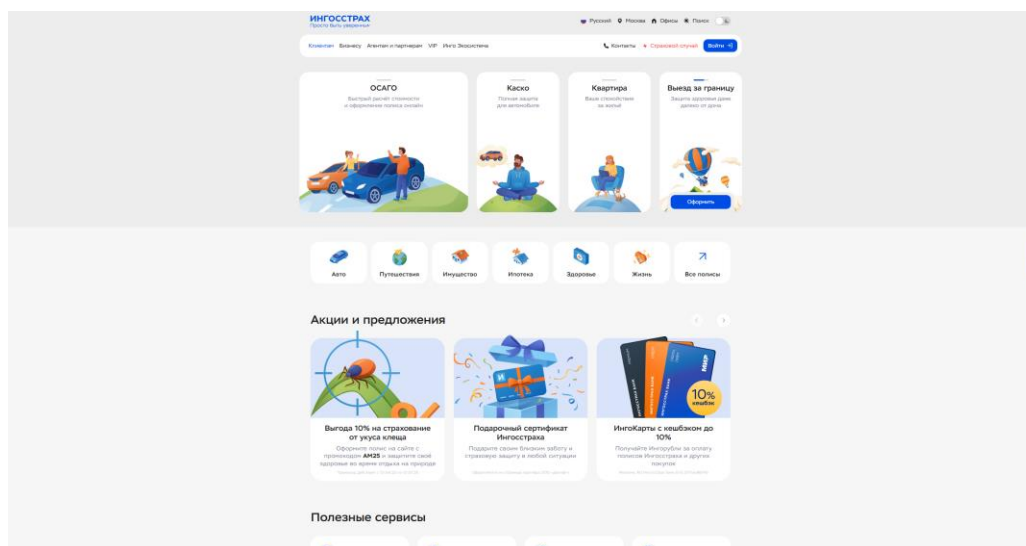


Рисунок 4 – Главная страница сайта

Для улучшения платформы необходимо оптимизировать производительность, доработать адаптивность. Эти изменения позволят сделать сайт более удобным и надежным для пользователей.

Веб-представительство страховой компании «Согласие» демонстрирует ряд существенных ограничений как в функциональности, так и в удобстве использования. На сайте реализован базовый калькулятор страхования, однако возможности онлайн-оформления полисов практически отсутствуют – пользователи могут лишь предварительно заполнить заявку, после чего им требуется завершить оформление через другие каналы, например, по телефону или в офисе. Личный кабинет также имеет минимальный функционал, что затрудняет управление заказами и просмотр истории взаимодействий с компанией.

Интерфейс сайта не имеет современного дизайна, но это не усложняет навигацию и поиск нужной информации (рисунок 5). Адаптивность сайта минимальна – на мобильных устройствах многие элементы интерфейса отображаются некорректно или становятся недоступными для взаимодействия.

Производительность сайта не выдерживает высоких нагрузок – при интенсивном использовании калькулятора или выполнении запросов наблюдаются значительные задержки и ошибки сервера.

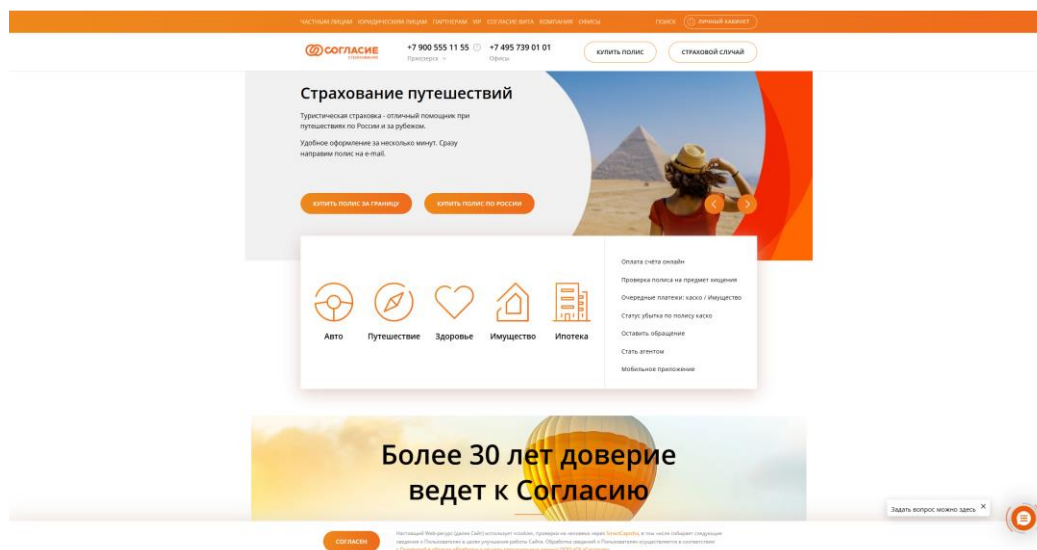


Рисунок 5 – Главная страница сайта

Кроме того, сайт не предлагает современных решений для улучшения пользовательского опыта, таких как интеграция с картами для поиска ближайших офисов или удобная фильтрация продуктов. Таким образом, веб-представительство компании «Согласие» требует значительной доработки, чтобы соответствовать современным стандартам удобства и производительности.

«Для анализа степени соответствия веб-представительства страхового агентства предъявляемым требованиям применяется таблица 2, а также установленные критерии оценки:

- полное несоответствие требованиям – 0 баллов;
- значительное несоответствие требованиям – 1 балл;
- незначительное несоответствие требованиям – 2 балла;
- полное соответствие требованиям – 3 балла» [8].

Таблица 2 – Сравнительный анализ аналогов веб-представительств страховых компаний

Критерий	Росгосстрах	Ингосстрах	Согласие
Наличие системы расчета стоимости полисов	3	3	3
Возможность просматривать все страховые продукты	3	3	2
Каждый продукт имеет собственную страницу с подробной информацией и ценами	3	2	2
Адаптивность и удобство использования на различных устройствах	3	2	0
Интуитивная навигация и дружелюбный интерфейс	2	1	0
Итого (макс. 15)	14	11	7

На основе проведенного анализа функциональных, технических и пользовательских требований к веб-представительствам страховых компаний, а также выявленных особенностей существующих аналогов, можно сделать вывод о необходимости комплексного подхода к разработке эффективного веб-представительства страхового агентства. Исследование показало, что успешное создание подобных систем зависит от учета множества факторов, начиная от функциональности и заканчивая безопасностью и удобством использования.

Одним из ключевых аспектов разработки веб-представительства является обеспечение его функциональной полноты, которая должна соответствовать ожиданиям как конечных пользователей, так и администраторов системы. Для достижения этих целей необходимо внедрить следующие функциональные компоненты:

- система расчета стоимости полисов – калькулятор должен быть интегрирован в интерфейс сайта и поддерживать все основные страховые продукты;

- онлайн-оформление заказов – возможность полностью оформить полис через сайт, включая заполнение необходимых данных, выбор способа оплаты и получение подтверждения;
- личный кабинет пользователя должен включать просмотр истории заказов и отслеживание статуса заявок;
- административная панель, позволяющая управлять страховыми продуктами, заказами, офисами и пользователями.

Анализ существующих веб-представительств страховых компаний показывает, что успешные сайты сочетают в себе удобный интерфейс, функциональность и информативность. «При разработке собственного веб-представительства важно учесть, как сильные стороны конкурентов, так и их недостатки, чтобы создать продукт, который будет максимально соответствовать потребностям пользователей» [13].

Выводы по главе 1

В рамках первой главы проведён всесторонний анализ особенностей разработки веб-представительства для страхового агентства. Определены ключевые характеристики предметной области и сформулированы основные требования, которым должна соответствовать проектируемая система. Исследование существующих аналогов показало различия в уровне их соответствия потребностям пользователей. На основе анализа были выделены основные требования – обеспечение высокой производительности, надежности и удобства использования, что направлено на повышение лояльности клиентов.

Таким образом, первая глава заложила теоретическую и методологическую основу для проектирования веб-представительства. Результаты исследования подтвердили актуальность темы и необходимость создания современного решения, отвечающего потребностям как пользователей, так и бизнеса.

Глава 2 Проектирование компонентов веб-представительства страхового агентства

2.1 Разработка логической модели веб-представительства

«Логическая модель веб-представительства страхового агентства представляет собой абстрактное описание структуры системы, включающее основные компоненты, их взаимодействие и функциональность. Эта модель разрабатывается на этапе проектирования и не зависит от конкретных технологий реализации. Она служит основой для дальнейшей разработки программного обеспечения и позволяет заранее выявить потенциальные проблемы, упрощает согласование требований с заказчиком и формирует понимание архитектуры будущего приложения» [8].

Для разработки логической модели веб-представительства страхового агентства, как правило, используется UML (Unified Modeling Language) – унифицированный язык моделирования. UML предоставляет стандартные графические обозначения для описания структуры, поведения и взаимодействия компонентов системы. Это позволяет создать четкое и понятное описание архитектуры приложения, которое не зависит от конкретных технологий реализации [1][5].

Для наглядности можно представить основные действия пользователей и администраторов в виде диаграммы вариантов использования (Use Case Diagram) (рисунок 6). Диаграммы вариантов использования показывают взаимодействие пользователей (актеров) с системой. Например, действия обычного пользователя – просмотр каталога, расчет стоимости полиса, регистрация [5][12].



Рисунок 6 – Диаграмма вариантов использования веб-представительства

Основные акторы (действующие лица):

Обычный пользователь – посетитель сайта, который хочет ознакомиться с информацией о страховых продуктах, рассчитать стоимость полиса или оформить заявку.

Доступные действия для обычного пользователя:

- просмотр каталога страховых продуктов;
- расчет стоимости полиса при помощи калькулятора;
- регистрация и авторизация на сайте;
- оформление заявки на страховой продукт;
- просмотр истории своих заказов в личном кабинете.

Администратор – пользователь с расширенными правами, который управляет данными системы (продукты, заказы, офисы).

Доступные действия для администратора:

- управление страховыми продуктами (добавление, редактирование, удаление);

- управление заказами (просмотр, изменение статуса);
- управление офисами (добавление, редактирование, удаление);
- управление пользователями (изменение ролей).

2) Варианты использования предоставлены в таблице 3.

Таблица 3 – Варианты использования управления системой

Участник	Название функции	Описание	Предусловия	Постусловия
Обычный пользователь	Просмотр страховых продуктов	Пользователь может просматривать список доступных страховых продуктов, их описание и стоимость.	Нет	Пользователь получает информацию о продуктах
Обычный пользователь	Расчет стоимости полиса	Пользователь выбирает страховой продукт и указывает сумму страхования, после чего система рассчитывает стоимость полиса.	Выбор продукта и ввод суммы страхования	Система возвращает расчетную стоимость полиса
Обычный пользователь	Оформление заявки	Пользователь оформляет заявку на выбранный страховой продукт, указывая сумму страхования.	Пользователь должен быть зарегистрирован и авторизован	Заявка сохраняется в базе данных
Обычный пользователь	Регистрация и авторизация	Пользователь регистрируется на сайте или авторизуется для доступа к личному кабинету.	Нет	Пользователь получает доступ к личному кабинету
Обычный пользователь	Просмотр личного кабинета	Пользователь может просматривать историю своих заказов и управлять личными данными.	Пользователь должен быть авторизован	Пользователь видит свои данные и историю заказов

Продолжение таблицы 3

Участник	Название функции	Описание	Предусловия	Постусловия
Администратор	Управление страховыми продуктами	Администратор добавляет, редактирует или удаляет страховые продукты.	Администратор должен быть авторизован	Изменения сохраняются в базе данных
Администратор	Управление заказами	Администратор просматривает список заказов и изменяет их статус (например, «обработан», «завершен»).	Администратор должен быть авторизован	Статус заказа обновляется
Администратор	Управление офисами	Администратор добавляет, редактирует или удаляет офисы страхового агентства.	Администратор должен быть авторизован	Информация об офисах обновляется в БД
Администратор	Управление пользователями	Администратор просматривает список пользователей и изменяет их роли (например, назначает роль «администратор»).	Администратор должен быть авторизован	Роль пользователя обновляется

Диаграмма последовательности показывает взаимодействие между объектами системы во времени. Например, рассмотрим процесс оформления заказа (рисунок 7) [6]:

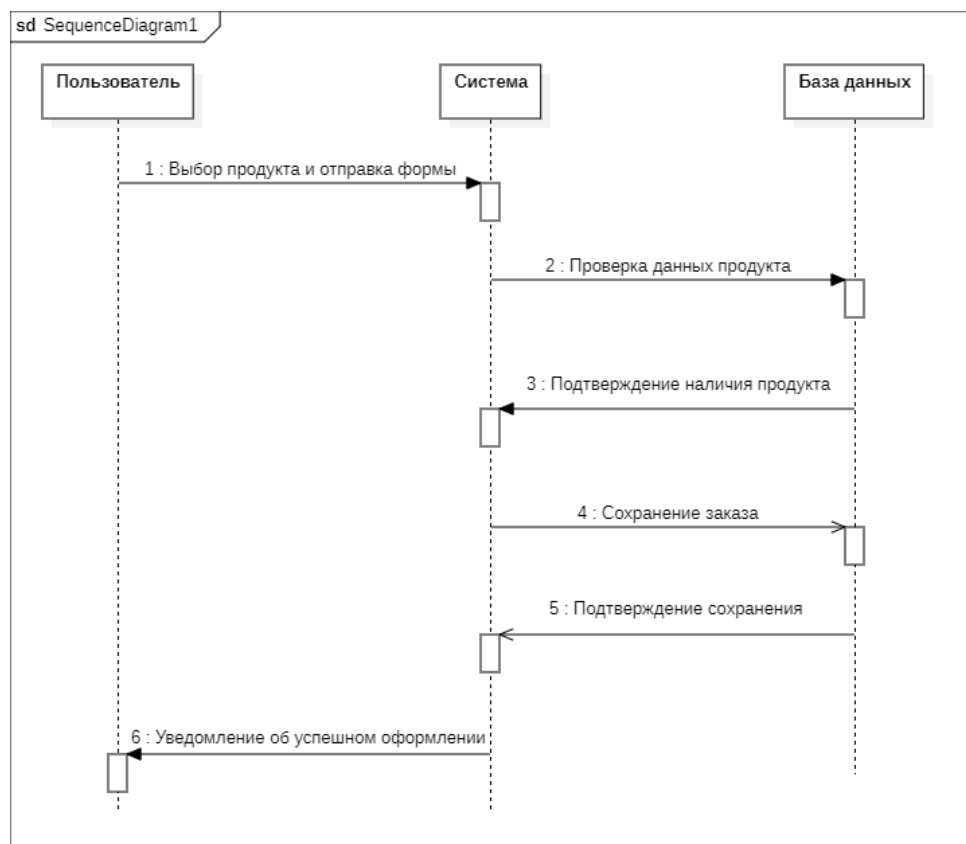


Рисунок 7 – Диаграмма последовательности веб-представительства

Пользователь выбирает страховой продукт и заполняет форму. Система проверяет данные и рассчитывает стоимость. Заказ сохраняется в базе данных. Пользователь получает уведомление об успешном оформлении.

«Диаграмма классов является одной из ключевых диаграмм в UML и используется для описания структуры системы на уровне объектно-ориентированного проектирования. Она показывает классы, их атрибуты, методы и взаимосвязи между ними. Для веб-представительства страхового агентства диаграмма классов помогает формализовать архитектуру базы данных и логику работы приложения» [4].

«На основании требования к функциональности веб-представительства страхового агентства можно выделить основные классы и их взаимосвязи, которые проставлены на рисунке 8» [1].

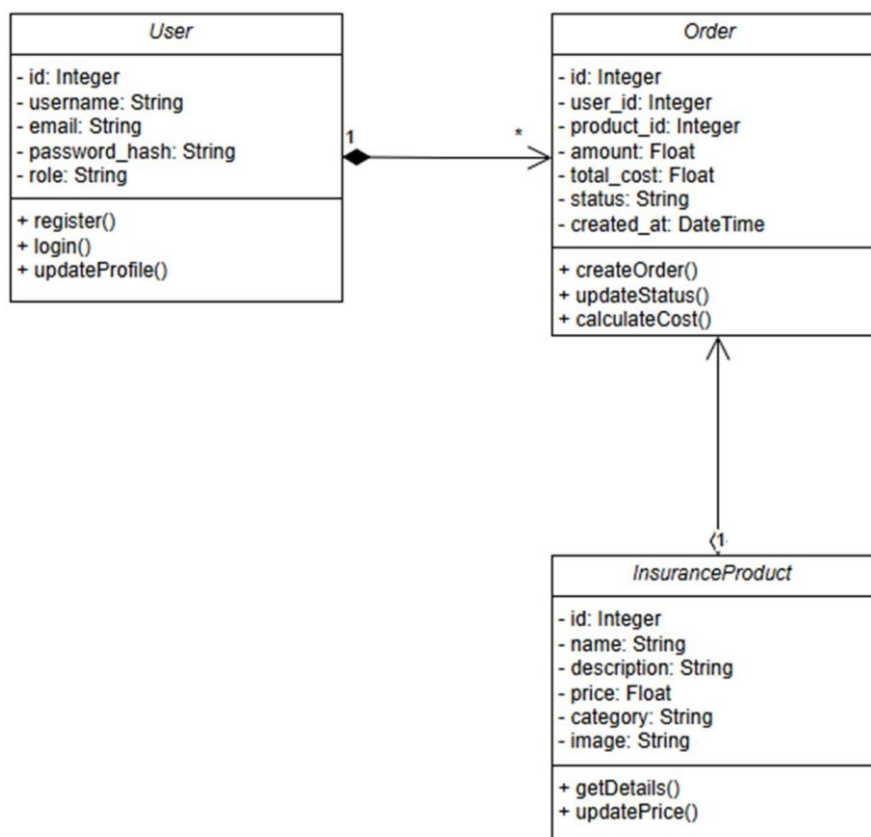


Рисунок 8 – Диаграмма классов веб-представительства

Класс **User** описывает зарегистрированных пользователей системы. Это может быть как обычный клиент, так и администратор. Этот класс управляет процессами регистрации, авторизации и обновления данных пользователей. Он также определяет роль пользователя, которая влияет на доступ к функциям системы. Имеет взаимосвязь с классом **Order**. Один пользователь может создать много заказов, но каждый заказ принадлежит только одному пользователю.

Класс InsuranceProduct описывает страховые продукты, предлагаемые агентством (например, ОСАГО, КАСКО, страхование жизни). Этот класс управляет данными о страховых продуктах, позволяя добавлять, редактировать и получать информацию о них. Он также используется для расчета стоимости полисов. Имеет взаимосвязь с классом Order. Один страховой продукт может быть выбран для многих заказов, но каждый заказ связан только с одним продуктом.

Класс Order описывает заявки на страховые продукты, созданные пользователями. Этот класс управляет жизненным циклом заказов – создание, расчет стоимости, изменение статуса и отслеживание. Он связывает пользователей и страховые продукты.

Класс Office описывает офисы страхового агентства, включая их местоположение. Этот класс предоставляет информацию об офисах компании, включая их расположение на карте и не имеет связей с другими классами. Он используется для отображения контактов и навигации.

«Диаграмма классов является важным инструментом для проектирования программного обеспечения. Она помогает формализовать структуру системы, определить взаимосвязи между компонентами и выявить потенциальные проблемы на этапе разработки. Для страхового агентства такая диаграмма служит основой для создания удобной, масштабируемой и надежной информационной системы» [12][14].

2.2 Логическая модель данных веб-представительства

«Логическая модель данных описывает структуру и взаимосвязи между сущностями (таблицами) в базе данных, которые используются для реализации функционала веб-представительства страхового агентства. Она фокусируется на

том, какие данные хранятся, как они организованы и как связаны между собой» [1].

Основные сущности (таблицы) представлены на рисунке 9:

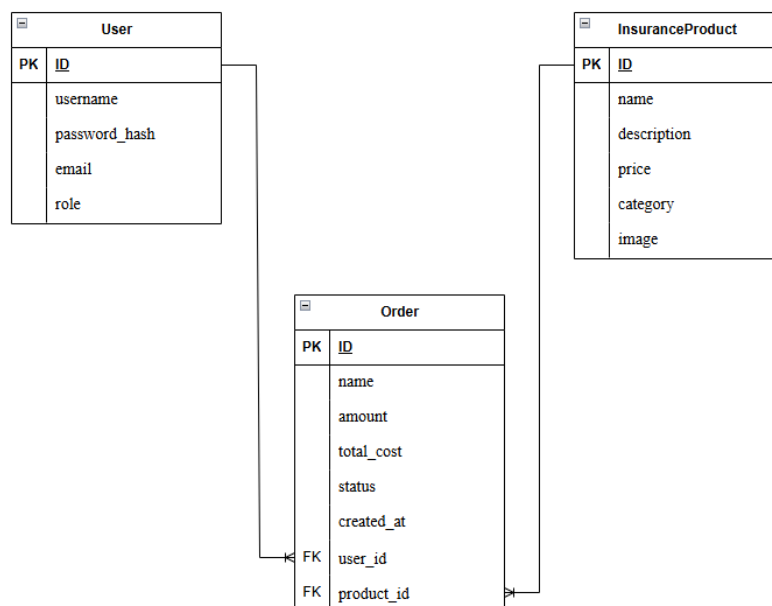


Рисунок 9 – Логическая модель данных веб-представительства

Пользователи (User) – представляет зарегистрированных пользователей системы.

«Поля:

- id (PK) – уникальный идентификатор пользователя;
- username – имя пользователя (логин);
- email – электронная почта пользователя;
- password_hash – хешированный пароль;
- role – роль пользователя (user, admin).

Взаимосвязи – один пользователь может иметь несколько заказов (Order).

Страховые продукты (InsuranceProduct) – представляет страховые продукты, предлагаемые агентством» [2].

«Поля:

- id (PK) – уникальный идентификатор продукта;
- name – название продукта;
- description – Описание продукта;
- price – стоимость продукта (в процентах от страховой суммы);
- category – категория продукта (например, «Автострахование», «Страхование жизни»);
- image – путь к изображению продукта» [2].

Взаимосвязи – один продукт может быть связан с несколькими заказами (Order).

Заказы (Order) – представляет заявки на страховые продукты, созданные пользователями.

«Поля:

- id (PK) – уникальный идентификатор заказа;
- user_id (FK) – идентификатор пользователя, создавшего заказ;
- product_id (FK) – идентификатор страхового продукта;
- amount – страховая сумма;
- total_cost – общая стоимость полиса;
- status – статус заказа (new, processed, completed);
- created_at – дата и время создания заказа» [2].

Взаимосвязи – связан с таблицами User и InsuranceProduct.

Описание взаимосвязей.

Связь между User и Order:

- один пользователь может создать много заказов;
- один заказ принадлежит одному пользователю.

Тип связи – один-ко-многим.

Связь между InsuranceProduct и Order:

- один страховой продукт может быть выбран для многих заказов;
- один заказ связан с одним страховым продуктом.

Тип связи – один-ко-многим.

«Ограничения целостности данных:

- первичные ключи (Primary Keys) – каждая таблица имеет уникальный первичный ключ (id), который гарантирует уникальность записей;
- внешние ключи (Foreign Keys) – в таблице Order: user_id ссылается на id в таблице User, product_id ссылается на id в таблице InsuranceProduct;
- проверка значений – поле price в таблице InsuranceProduct должно быть положительным числом. Поле status в таблице Order должно принимать значения из ограниченного набора (new, processed, completed)» [2].

Таким образом, представленная логическая модель данных обеспечивает структурированное хранение информации и определяет четкие взаимосвязи между основными сущностями веб-представительства страхового агентства. Она позволяет эффективно управлять пользователями, страховыми продуктами, заказами, обеспечивая тем самым надежную основу для реализации функциональных возможностей системы. Внедрение ограничений целостности данных гарантирует корректность и согласованность информации [12].

2.3 Разработка структуры веб-представительства страхового агентства

«Создание сайта или любого другого онлайн-ресурса начинается с формирования его структуры – логической схемы, отражающей взаимосвязь между страницами и разделами. Этот этап играет центральную роль, поскольку напрямую влияет на удобство перемещения по сайту, функциональность платформы и организацию контента. Грамотно продуманная структура не только

обеспечивает комфортное взаимодействие пользователей с ресурсом, но и способствует успешному продвижению в поисковых системах, делая сайт более доступным для целевой аудитории» [2].

Структура сайта –это основа, на которой строится весь проект. Она влияет на:

- пользовательский опыт (UX) – четкая организация разделов позволяет пользователям быстро находить нужную информацию, что повышает их удовлетворенность и снижает процент отказов;
- поисковую оптимизацию (SEO) – поисковые системы лучше индексируют сайты с логичной иерархией страниц и корректными URL;
- масштабируемость – грамотно спроектированная структура упрощает добавление новых разделов или страниц без нарушения общей логики;
- производительность – оптимизация структуры помогает уменьшить время загрузки страниц за счет минимизации количества запросов к серверу.

«Так же структура может быть двух видов – внешней и внутренней.

Внешняя, которую визуально видят пользователи ресурса, может включать в себя следующие элементы:

- хедер, шапка – блок в верхней части ресурса с названием, логотипом, фото продукта. Здесь же часто размещается баннерная реклама;
- сайдбар – боковая панель, располагающаяся в правой или левой части страницы. Sidebar предназначен для дополнительной информации, которая привлекает внимание посетителя и мотивирует его дольше задержаться на сайте. Это могут быть, к примеру, анонсы интересных материалов или акций, нового контента;

- центральный блок – это основные данные в центре страницы, за которыми посетитель зашел на сайт. Она должна быть хорошо структурированной и заметной, иметь заголовки;
- подвал (футер) – нижняя часть сайта, где обычно располагаются контакты, телефон, адрес и полезные ссылки для быстрой навигации (например, связь с технической поддержкой)» [2].

С учетом функциональных требований предлагается следующая организация разделов:

- главная страница – общая информация о компании, преимущества и краткий список страховых услуг;
- страховые продукты (каталог) – список доступных типов страхования (медицинское, автострахование, страхование путешествий);
- онлайн-калькулятор – интерфейс для расчета предварительной стоимости полиса с учетом базового тарифа и выбранных параметров;
- личный кабинет (доступен только зарегистрированным пользователям)
 - просмотр оформленных полисов, статуса оплат, персональных данных;
- контакты – контактная информация, адреса офисов.

Для удобства пользователей целесообразно расположить глобальное меню навигации в верхней части страницы (header), включив туда ссылки: «Главная», «Услуги/Продукты», «Калькулятор», «Личный кабинет» (если пользователь авторизован), «Контакты».

Требования к навигации:

- простота – минимальное количество кликов до нужной информации;
- единый стиль – все страницы оформляются в одном стиле и используют одну структуру меню;

- адаптивность – корректное отображение навигации на мобильных устройствах (бургер-меню).

«Внутренняя структура – это структура для разработчиков продукта, она состоит из категорий и материалов, связанных между собой. Внутренняя структура может быть:

- линейная, пользователь просматривает поочередно каждую страницу. Страницы одновременно ссылаются и на главную страницу, и на друг друга. Подобная структура подходит для мероприятий, презентаций или портфолио;
- линейная с ответвлениями – тип структуры, очень похожий на линейную, но с разницей в том, что страницы объединены в блоки;
- блочная структура подразумевает, что каждая страница ссылается на несколько других, равнозначных между собой. Такая структура хорошо подходит для представления товара, где на страницах будут описания отдельно взятых свойств продукта;
- древовидная – самая универсальная структура. В данной структуре каждая страница располагается в отдельной ветке, которая имеет вид раздела. В удобных древовидных структурах содержание контента распределяется между разделами и главной страницей, поскольку все они ссылаются друг на друга» [2].

Создание четкой и логичной структуры сайта является ключевым этапом разработки, влияющим на удобство использования, функциональность и SEO-эффективность. Внешняя структура, должна обеспечивать удобную навигацию и доступ к основным разделам, таким как каталог страховых продуктов, онлайн-калькулятор и личный кабинет. Внутренняя структура для страхового агентства предпочтительна древовидная. Особое внимание следует уделить простоте навигации, адаптивности и единому стилю, чтобы обеспечить комфортное

взаимодействие пользователей и поддерживать актуальность ресурса для будущего расширения функционала [2][13][15].

Выводы по главе 2

Логическая модель веб-представительства страхового агентства была разработана в ходе проектирования и включает описание ключевых компонентов системы, их взаимодействия и реализуемых функций. Данная модель имеет абстрактный характер и не привязана к конкретным технологическим решениям, что обеспечивает её универсальность и применимость в качестве основы для дальнейшего этапа проектирования и реализации программного обеспечения.

На основе анализа требований выбрана внутренняя архитектура приложения, оптимально соответствующая целям и задачам разрабатываемого веб-ресурса.

Также спроектирована внешняя структура сайта – определена система навигации и выделены основные разделы, ориентированные на удобство использования и обеспечение эффективного доступа пользователей к предоставляемым услугам.

Глава 3 Реализация веб-представительства страхового агентства

3.1 Разработка графического дизайна веб-представительства

«Графический дизайн веб-представительства играет ключевую роль в формировании первого впечатления у пользователей и определяет удобство взаимодействия с платформой. Он включает в себя не только эстетическую составляющую, но и функциональность, доступность и соответствие требованиям целевой аудитории» [3].

Дизайн веб-представительства может включать в себя набор из следующих элементов:

- логотип – узнаваемое графическое изображение, благодаря которому пользователь узнает бренд или продукт;
- фирменный стиль – включает в себя набор элементов, которые объединяют все продукты и коммуникации в единое целое (корпоративная цветовая палитра, шрифты и так далее);
- различные блоки – это внешняя структура сайта, шапка, подвал, центральный блок.

Прототип (wireframe) главной страницы (рисунок 10), демонстрирующий расположение ключевых элементов.

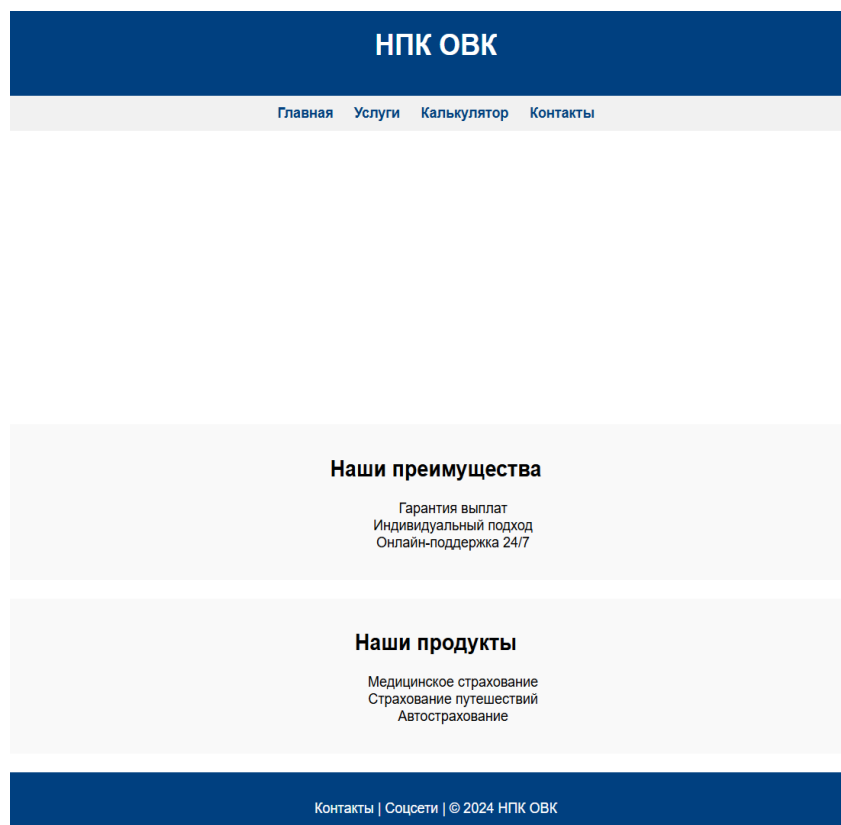


Рисунок 10 – Прототип (wireframe) главной страницы

Для страхового агентства обычно подходит сдержанная цветовая палитра, вызывающая доверие и подчеркивающая серьезность бренда. Рекомендуются спокойные оттенки синего, зеленого или серого цветов в сочетании с белым фоном и контрастными акцентами. Например:

- основной цвет – темно-синий (#1A3D5D);
- дополнительный цвет – светло-серый (#F2F2F2);
- цвет акцентов – ярко-синий (#2F80ED) или зеленоватый оттенок (#2DCB73).

При адаптивной верстке необходимо следить, чтобы сайт корректно отображался на различных разрешениях экрана: мобильных телефонах, планшетах, ноутбуках.

Прототип главной страницы демонстрирует логичное расположение элементов для удобства навигации, а адаптивная верстка с использованием современных технологий обеспечивает корректное отображение на всех устройствах. Грамотный дизайн повышает пользовательский опыт, лояльность клиентов и конкурентоспособность бренда.

3.2. Обоснование архитектуры веб-представительства

Архитектура веб-представительства страхового агентства разработана с учетом требований к функциональности, масштабируемости, безопасности и к удобству поддержки. Ниже приведено детальное обоснование выбора архитектурных решений.

Для реализации веб-представительства была выбрана архитектура «клиент-сервер», которая является стандартом для современных веб-приложений. Она обеспечивает четкое разделение логики работы клиента (браузера) и сервера:

- клиентская часть отвечает за отображение интерфейса пользователя и взаимодействие с сервером через HTTP-запросы;
- используются HTML, CSS и JavaScript (включая библиотеки, такие как Bootstrap, для адаптивного дизайна);
- формы и элементы интерфейса позволяют пользователям выполнять действия – регистрацию, авторизацию, расчет стоимости полиса, оформление заказов и так далее [21].

Серверная часть – обрабатывает запросы от клиентской части, выполняет бизнес-логику приложения и взаимодействует с базой данных. Серверная часть реализована на Python с использованием фреймворка Flask. Flask был выбран за его легковесность, гибкость и простоту настройки [18][19].

Для организации кода была выбрана модель MVC (Model-View-Controller), которая обеспечивает разделение ответственности между компонентами приложения:

- Model (Модель) - представляет данные и бизнес-логику приложения. Модели определены в SQLAlchemy (ORM – объектно-реляционное отображение), что позволяет работать с базой данных через объекты Python. Примеры моделей: User, InsuranceProduct, Order, Office;
- View (Представление) - отвечает за отображение данных пользователю. Шаблоны HTML рендерятся с использованием Jinja2 (движок шаблонов Flask). Каждый шаблон соответствует определенной странице или компоненту интерфейса;
- Controller (Контроллер) – обрабатывает запросы от клиента, взаимодействует с моделями и возвращает представления. Маршруты (routes) Flask выступают в роли контроллеров. Например, маршрут /submit_order обрабатывает POST-запросы для создания новых заказов.

Проект организован в виде модульной структуры, что упрощает поддержку и расширение (рисунок 11).

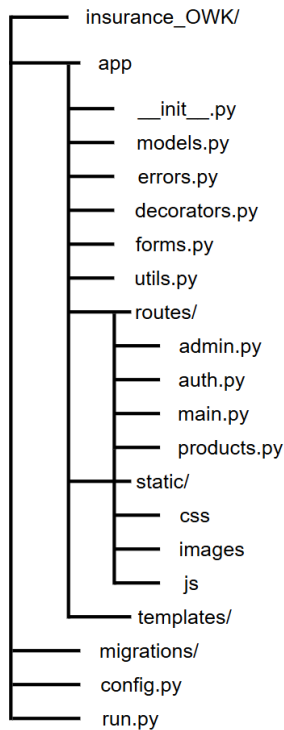


Рисунок 11 – Структура проекта

Для хранения данных используется SQLAlchemy – ORM (объектно-реляционное отображение), которое предоставляет следующие преимущества:

- удобство работы с базой данных через Python-объекты;
- поддержка миграций через Flask-Migrate (Alembic);
- возможность использования различных СУБД (SQLite для разработки, PostgreSQL для выпуска).

Архитектура веб-представительства включает несколько уровней безопасности:

- аутентификация и авторизация реализована через Flask-Login. Пользователи могут регистрироваться и авторизовываться;
- администраторы имеют доступ к специальным маршрутам (панель управления);

- пароли хранятся в базе данных в зашифрованном виде с использованием `werkzeug.security.generate_password_hash`;
- Flask-WTF автоматически добавляет защиту от CSRF-атак в формы;
- декоратор `@login_required` ограничивает доступ к защищенным маршрутам только авторизованным пользователям;
- дополнительно проверяется роль пользователя (`is_admin`) для доступа к административным функциям.

Архитектура разработана с учетом возможности масштабирования:

- горизонтальное масштабирование – приложение может быть развернуто на нескольких серверах с использованием балансировки нагрузки (например, Nginx);
- интеграция с внешними сервисами – реализовано кодирование адресов через сторонний API (например, Yandex Maps или Google Maps);
- разделение на микросервисы – если потребуется, функционал можно разделить на микросервисы (например, отдельный сервис для расчета стоимости полиса).

Для обеспечения высокой производительности:

- кэширование – часто запрашиваемые данные (например, список страховых продуктов) могут быть закэшированы с использованием Redis или Memcached;
- оптимизация запросов – используются индексы в базе данных для ускорения поиска. Запросы оптимизированы для минимизации времени выполнения;
- тестирование производительности – проведено нагрузочное тестирование с использованием JMeter для оценки производительности системы при большом количестве одновременных пользователей.

Кроссплатформенность и адаптивность:

- адаптивный дизайн – использование Bootstrap обеспечивает корректное отображение интерфейса на устройствах с различными разрешениями (десктопы, планшеты, смартфоны);
- поддержка различных браузеров – приложение протестировано в популярных браузерах (Chrome, Firefox, Safari, Edge).

Выбранная архитектура веб-представительства страхового агентства обеспечивает:

- гибкость и масштабируемость;
- безопасность и надежность;
- удобство разработки и поддержки;
- высокую производительность и кроссплатформенность.

Эти решения позволяют создать современное и эффективное веб-приложение, отвечающее требованиям как пользователей, так и администраторов [9][11].

3.3 Средства реализации веб-представительства

Для успешной разработки веб-представительства страхового агентства необходимо выбрать подходящие инструменты и технологии. Этот выбор основывается на требованиях проекта, таких как производительность, масштабируемость, удобство разработки и поддержка современных стандартов.

Основным языком программирования для реализации проекта выбран Python. Python является одним из самых популярных языков для веб-разработки благодаря своей простоте, читаемости кода и богатой экосистеме библиотек и фреймворков [10].

Для реализации серверной части веб-приложения выбран микрофреймворк Flask. Flask обладает следующими преимуществами:

- легковесность - Flask предоставляет минимальный набор функционала, что позволяет разработчику иметь полный контроль над структурой приложения;
- гибкость - возможность легко интегрировать сторонние библиотеки (например, SQLAlchemy для работы с базой данных, Flask-Login для управления пользователями);
- масштабируемость - Flask подходит как для небольших проектов, так и для крупных приложений с использованием Blueprint'ов;
- активное сообщество - большое количество документации, примеров и готовых решений.

Для создания пользовательского интерфейса используются стандартные веб-технологии:

- «HTML – структура страниц;
- CSS – оформление и дизайн;
- JavaScript – динамическое взаимодействие с пользователем;
- Bootstrap – фреймворк для быстрой разработки адаптивного дизайна;
- Bootstrap Icons – иконки для улучшения визуального восприятия;
- Jinja2 – шаблонизатор Flask для динамической генерации HTML» [9].

Для хранения данных выбрана реляционная система управления базами данных SQLite. SQLite подходит для проектов начального уровня благодаря следующим характеристикам:

- простота использования – база данных хранится в одном файле, что упрощает развертывание и управление;
- безопасность – поддержка транзакций и целостности данных;
- производительность – хорошо работает с небольшими и средними объемами данных.

Для взаимодействия с базой данных используется ORM SQLAlchemy, который предоставляет следующие преимущества:

- удобный синтаксис для работы с базой данных;
- защита от SQL-инъекций;
- поддержка миграций через Flask-Migrate.

Выбранные средства реализации обеспечивают быструю разработку и тестирование, гибкость и масштабируемость, современный и удобный пользовательский интерфейс. Эти технологии позволяют создать надежное и эффективное веб-представительство страхового агентства, соответствующее всем требованиям проекта.

3.4 Физическая модель данных веб-представительства

«Физическая модель данных описывает, как данные будут храниться в базе данных на уровне реализации. Она включает структуру таблиц, типы данных, первичные и внешние ключи, индексы и другие аспекты, связанные с физическим хранением данных. В данном разделе мы детализируем физическую модель данных для веб-представительства страхового агентства» [7].

База данных состоит из нескольких таблиц, каждая из которых представляет определенный объект предметной области:

- «таблица пользователей системы (user) содержит Уникальный идентификатор пользователя (Первичный ключ, целое число, не пустое значение), имя пользователя (текстовое уникальное поле) электронная почта (текстовое уникальное поле), хешированный пароль (текст), роль пользователя (текстовое поле с определенным значением «user» или «admin»);
- таблица продуктов (insurance_product) содержит Уникальный идентификатор продукта (Первичный ключ, целое число, не пустое

значение), название продукта (текстовое поле), описание продукта (текстовое поле), цена продукта (в процентах, дробное число), категория продукта (текстовое поле) и путь к изображению продукта (текстовое поле);

- таблица заказов (order) содержит Уникальный идентификатор заказа (Первичный ключ, целое число, не пустое значение), сумма страхования (дробное число), итоговая стоимость полиса (дробное число), статус заказа (текстовое поле со значениями «new», «processed», «completed»), дата и время создания заказа (дата и время), идентификатор пользователя (внешний ключ, ссылается на значение УИД таблицы user, целое число, не пустое значение), идентификатор страхового продукта (внешний ключ, ссылается на значение УИД таблицы insurance_products, целое число, не пустое значение);
- таблица офисов компании (office) содержит Уникальный идентификатор офиса (Первичный ключ, целое число, не пустое значение), название офиса (текстовое поле), адрес офиса (текстовое поле), широта (географические координаты, дробное число), долгота (географические координаты, дробное число).

Связь между user и order реализована через внешний ключ user_id в таблице order. Один пользователь может иметь несколько заказов, что указывает на отношение «один-ко-многим» (1:N).

Связь между insurance_product и order реализована через внешний ключ product_id в таблице order. Один страховой продукт может быть связан с несколькими заказами, что указывает на отношение «один-ко-многим» (1:N)» [20].

Для улучшения производительности запросов созданы следующие индексы:

- индекс по полю email в таблице users: для быстрого поиска пользователей по электронной почте;
- индекс по полю name в таблице insurance_products: для быстрого поиска продуктов по названию;
- составной индекс по полям user_id и status в таблице orders: для фильтрации заказов по пользователю и статусу.

Для оптимизации хранения данных таблицы спроектированы в соответствии с принципами нормализации (до третьей нормальной формы), чтобы избежать дублирования данных и обеспечить целостность.

Хранение изображений страховых продуктов осуществлено в файловой системе, а в базе данных сохраняются только пути к файлам. Это позволяет минимизировать нагрузку на базу данных.

Географические координаты офисов (latitude и longitude) используются для интеграции с картами. Для их хранения выбран тип данных FLOAT, что обеспечивает точность до шести знаков после запятой.

Физическая модель данных разработана с учетом требований предметной области и особенностей реализации. Она обеспечивает:

- эффективное хранение данных;
- быстрый доступ к информации;
- возможность масштабирования системы.

Эта модель является основой для реализации функционала Веб-представительства страхового агентства.

3.5 Маршрутизация веб-страниц приложения

Маршрутизация веб-страниц является важным аспектом разработки веб-приложений, так как она определяет, как пользователи взаимодействуют с различными частями системы. В контексте реализации Веб-представительства

страхового агентства маршрутизация была организована с учетом удобства навигации, логической структуры приложения и требований к производительности.

При разработке маршрутизации были рассмотрены два основных подхода:

- абсолютный путь отражает полную структуру URL, включая все предыдущие сегменты. Пример абсолютного пути `/products/view/1` (просмотр продукта с ID 1). Преимущества данного подхода заключается в четкой иерархии URL, что упрощает понимание структуры сайта. Полезен для SEO (поисковая оптимизация), так как длинные URL лучше отражают содержимое страницы. Недостатком является длина URL. URL могут быть менее удобны для восприятия пользователями;
- относительный путь обеспечивает доступ к странице независимо от текущего положения пользователя на сайте. Пример относительного пути `view-product/1` (просмотр продукта с ID 1). Преимущества данного подхода заключается в упрощенных URL, которые легче запомнить и ввести. Более гибкий доступ к страницам без привязки к контексту. Недостатком является меньшая информативность для поисковых систем и пользователей.

Таким образом маршрутизация веб-страниц в веб-представительстве страхового агентства была организована с учетом удобства пользователей, логической структуры приложения и требований к производительности. Использование Flask позволило создать гибкую и масштабируемую систему маршрутов, которая легко поддерживается и расширяется [9][11].

3.6 Проектирование и разработка пользовательского интерфейса

«Пользовательский интерфейс (UI) является ключевым компонентом любого программного обеспечения, так как он определяет, насколько удобно и интуитивно пользователи могут взаимодействовать с системой. В рамках данного проекта проектирование и разработка интерфейса веб-представительства страхового агентства были выполнены с учетом требований удобства, доступности и функциональности» [2].

При разработке интерфейса были учтены следующие принципы:

- интуитивная понятность – интерфейс должен быть простым и понятным для пользователей с различным уровнем технической подготовки. Это достигается за счет использования знакомых паттернов (например, кнопки «Войти», «Зарегистрироваться») и минимизации количества шагов для выполнения задач;
- доступность – интерфейс адаптирован для пользователей с ограниченными возможностями. Для этого использованы контрастные цвета, крупные шрифты и поддержка навигации с клавиатуры (например, табуляция);
- адаптивность – интерфейс корректно отображается на устройствах с различными размерами экранов (настольные компьютеры, планшеты, смартфоны). Это достигается за счет использования адаптивного дизайна и медиа-запросов в CSS;
- Привлекательный дизайн – использована современная цветовая схема и графические элементы, которые делают интерфейс визуально привлекательным, но не перегружают пользователя лишними деталями;
- скорость работы – интерфейс разработан с учетом быстрого отклика системы. Например, форма расчета стоимости полиса обрабатывается

без необходимости перезагрузки страницы благодаря использованию AJAX.

Основные страницы интерфейса:

- главная страница предоставляет ссылки на разделы сайта отображает информацию о страховых продуктах;
- страница регистрации и авторизации содержит поля для ввода логина, пароля и электронной почты, а также кнопки «Зарегистрироваться» и «Войти»;
- личный кабинет пользователя содержит информацию об истории заказов и уведомления о статусе заявок;
- административная панель предоставляет возможность управления страховыми продуктами (добавление, редактирование, удаление), просмотр и изменение статуса заказов, управление офисами и пользователями. Пример программного кода добавления страхового продукта показан на рисунке 12:

```

@admin_bp.route('/admin/add_product', methods=['GET', 'POST'])
@login_required
def add_product():
    if not current_user.is_admin():
        flash('У вас нет прав для доступа к этой странице!', 'danger')
        return redirect(url_for('main.index'))

    if request.method == 'POST':
        name = request.form['name']
        description = request.form['description']
        price = float(request.form['price'])
        category = request.form['category']

        # Обработка загрузки изображения
        image_url = None
        if 'image' in request.files:
            file = request.files['image']
            if file and allowed_file(file.filename):
                # Убедитесь, что директория существует
                if not os.path.exists(current_app.config['UPLOAD_FOLDER']):
                    os.makedirs(current_app.config['UPLOAD_FOLDER'])

                filename = secure_filename(file.filename)
                file_path = os.path.join(current_app.config['UPLOAD_FOLDER'], filename)
                file.save(file_path)
                image_url = f"/static/images/{filename}"

        new_product = InsuranceProduct(
            name=name,
            description=description,
            price=price,
            category=category,
            image=image_url
        )
        db.session.add(new_product)
        db.session.commit()

        flash('Страховой продукт успешно добавлен!', 'success')
        return redirect(url_for('admin.admin_panel'))

    return render_template('add_product.html')

```

Рисунок 12 – Пример программного кода добавления страхового продукта

Готовый интерфейс страницы административной панели показан на рисунке 13:

Панель администратора

Страхование продукты

Добавить продукт

ID	Название	Описание	Цена (%)	Категория	Действия
1	ОСАГО	Обязательное автострахование	5.0%	Автострахование	Редактировать Удалить
2	КАСКО	Добровольное автострахование	10.0%	Автострахование	Редактировать Удалить
3	Страхование жизни	Страхование здоровья и жизни	3.0%	Страхование жизни	Редактировать Удалить

Заказы пользователей

ID	Пользователь	Продукт	Сумма страхования	Итоговая стоимость	Статус	Дата создания	Действия
1	123	ОСАГО	20000.0 руб.	1000.0 руб.	completed	2025-04-21 20:51	Завершен Сохранить
2	Вячеслав	КАСКО	10000.0 руб.	1000.0 руб.	new	2025-05-01 18:48	Новый Сохранить
3	Вячеслав	КАСКО	10000.0 руб.	1000.0 руб.	new	2025-05-01 18:51	Новый Сохранить

Управление пользователями

ID	Имя пользователя	Email	Текущая роль	Изменить роль
1	admin	admin@example.com	admin	Администратор Изменить
2	123	okolinvyacheslav@yandex.ru	admin	Администратор Изменить
3	ZAP	zaproxy@example.com	user	Пользователь Изменить
4	Вячеслав	qazex298@gmail.com	user	Пользователь Изменить

Офисы

Добавить офис

ID	Название	Адрес	Широта	Долгота	Действия
----	----------	-------	--------	---------	----------

Рисунок 13 – Панель администратора

Панель администратора интуитивно понятна. Для удобства восприятия кнопки редактирования, удаления и добавления выделены разными цветами.

В приложении А можно ознакомиться с другими основными страницами и формами веб-представительства.

3.7 Размещение веб-представительства на хостинге

При выборе хостинга для размещения веб-приложения важно учитывать такие факторы, как производительность, надежность, масштабируемость, удобство управления и стоимость. В данном случае рассматривается Cloud Apps

от Timeweb Cloud, который является одним из популярных решений для размещения веб-приложений на российском рынке.

Основные преимущества:

- автоматизация – упрощенное развертывание приложений с помощью предустановленных шаблонов (например, для Flask, Django, Node.js);
- масштабируемость – возможность увеличивать или уменьшать ресурсы (CPU, RAM, SSD) в зависимости от нагрузки;
- гибкость – поддержка различных языков программирования (Python, PHP, Node.js, Ruby и др.) и баз данных (PostgreSQL, MySQL, MongoDB);
- безопасность – защита данных с помощью SSL-сертификатов, бэкапов и DDoS-защиты;
- локализация – серверы находятся на территории России, что обеспечивает низкую задержку для пользователей из РФ.

Технические характеристики выбранного хостинга:

Процессор и память:

- CPU: от 1 до 8 ядер в зависимости от выбранного тарифа;
- RAM: от 1 ГБ до 16 ГБ;
- SSD-диски от 10 ГБ до 100 ГБ.

Базы данных:

- поддерживаемые СУБД: PostgreSQL, MySQL, MongoDB, Redis;
- автоматическое создание базы данных при развертывании приложения.

Языки программирования:

- Python (поддержка Flask, Django, FastAPI и других фреймворков);
- PHP (Laravel, Symfony);
- Node.js;
- Ruby on Rails.

Домены:

- поддержка собственных доменов;
- автоматическая привязка домена через dns-записи;
- бесплатное использование SSL-сертификатов Let's Encrypt.

Выбранная конфигурация для развертывания сайта выбрана на начальном этапе минимальная:

- CPU: 1x33ГГц;
- RAM: 1ГБ;
- дисковое пространство: 15ГБ;
- авторазвертывание: Включено;
- стоимость: 188 р./мес.;
- домен - okolindev-insurance-owk-9b2f.twc1.net.

Cloud Apps от Timeweb Cloud – это надежное и удобное решение для размещения веб-приложения страхового агентства. Оно предоставляет все необходимые инструменты для быстрого развертывания, масштабирования и управления приложением. Благодаря локализации, автоматизации и конкурентной цене, это оптимальный выбор для проектов, ориентированных на российскую аудиторию.

3.8 Тестирование Jmeter

Тестирование с использованием инструмента Apache JMeter позволяет оценить производительность веб-приложения под нагрузкой. Это особенно важно для веб-представительства страхового агентства, так как система должна быть способна обрабатывать большое количество одновременных запросов от пользователей.

Тестирование на отказоустойчивость проводилось в 200 потоков (200 пользователей) запущенных в промежутке 30 секунд, и каждый пользователь выполнял свои действия 50 раз.

Тестирование страницы /products (рисунок 14):

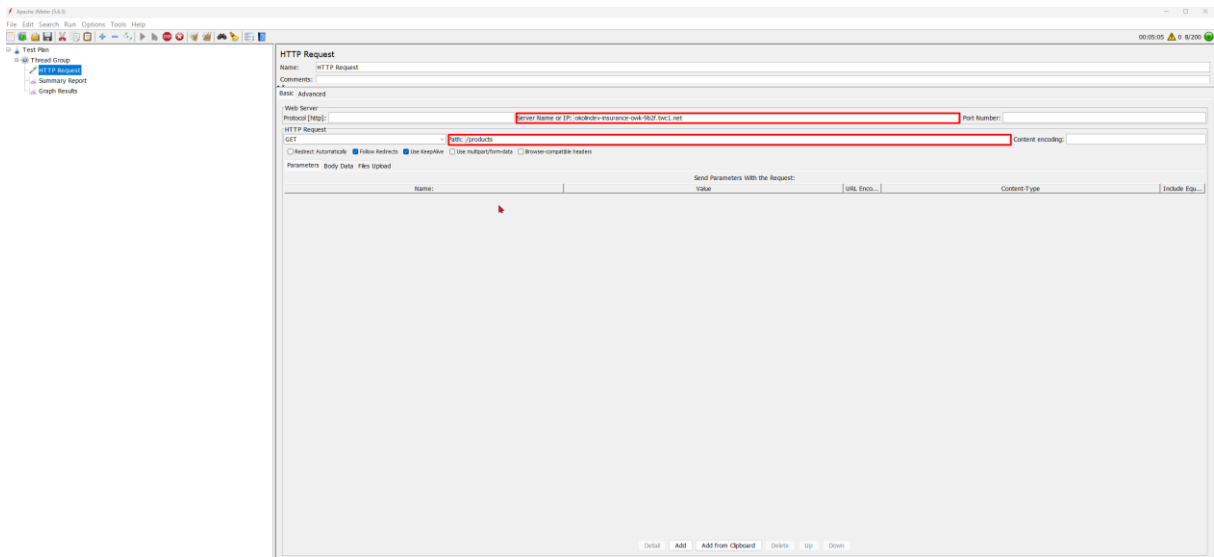


Рисунок 14 – Настройки Jmeter для тестирования страницы /products

Ниже предоставлен суммарный отчет результатов тестирования страницы продуктов (рисунок 15).

The screenshot shows the Apache JMeter GUI with the 'Summary Report' selected. The report displays the following data:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec
HTTP Request	10000	1425	0	121075	7346.36	0.50%	30.33/sec	238.12	8.2
TOTAL	10000	1425	0	121075	7346.36	0.50%	30.33/sec	238.12	8.2

Рисунок 15 – Суммарный отчет результатов тестирования

Ниже предоставлен отчет в виде графика (рисунок 16).



Рисунок 16 – Отчет о тестировании в виде графика.

Таким образом в ходе тестирования страницы продуктов было проведено 10 тысяч запросов. Процент ошибок 0,92%. Исходя из результатов тестирования можно сделать выводы, что тестирование на отказоустойчивость пройдено успешно. Сайт работает стабильно при больших нагрузках.

3.9 Тестирование ZAP 2.16.1

ZAP (Zed Attack Proxy) – это популярный инструмент для тестирования безопасности веб-приложений. Он позволяет выявлять уязвимости, такие как XSS (межсайтовый скриптинг), SQL-инъекции, CSRF (межсайтовая подделка запросов) и другие угрозы [16].

При тестировании приложения было выявлено отсутствие защиты от CSRF-атак (рисунок 17).

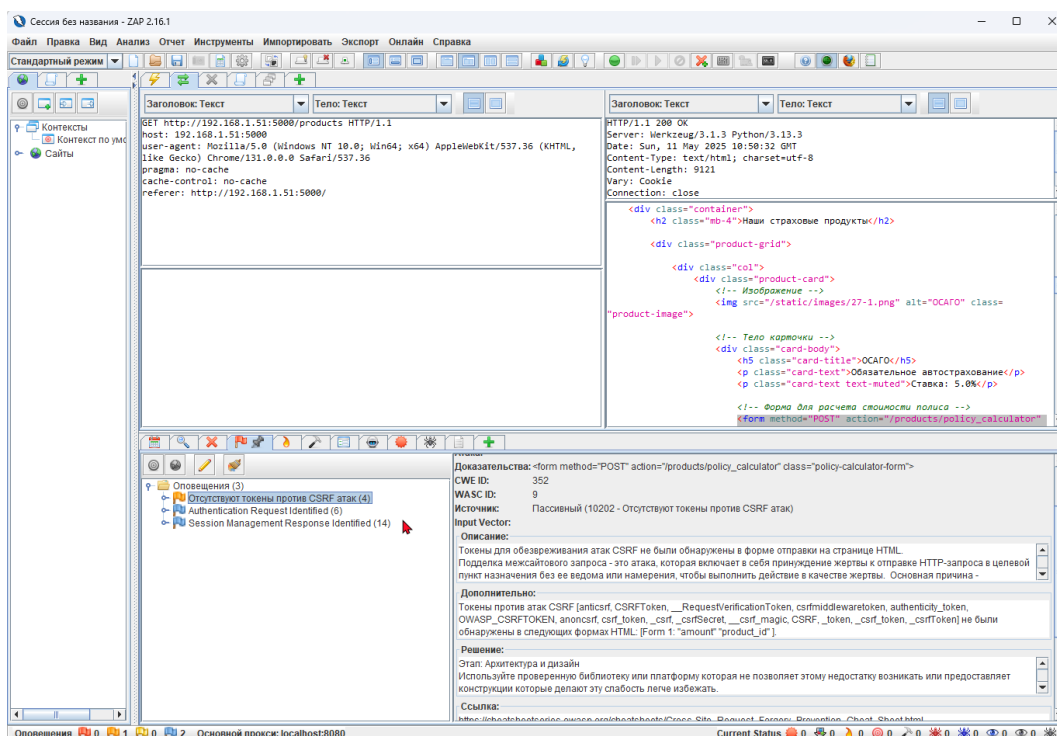


Рисунок 17 – Отсутствие защиты от CSRF-атак.

Для защиты от CSRF-атак необходимо добавить токены CSRF в формы и настроить серверную часть для проверки этих токенов. Для реализации этой защиты необходимо модифицировать HTML-код и Flask-приложение:

- добавление CSRF-токена в HTML-формы. В каждой форме, где выполняется POST-запрос, необходимо добавить скрытое поле с CSRF-токеном «`<input type="hidden" name="csrf_token" value="{{ csrf_token() }}">`». Это можно сделать с помощью Flask-WTF;
- необходимо настроить Flask-приложение, добавить импорт Flask-WTF CSRFProtect и включить защиту приложения `csrf = CSRFProtect(app)`.

Теперь приложение защищено от CSRF-атак. Все формы содержат CSRF-токены, а сервер проверяет их перед обработкой запросов. Это значительно повышает безопасность приложения.

Выводы по главе 3

Подготовлен базовый графический дизайн (цветовая палитра, прототипы ключевых страниц, элементы брэндинга), отражающий фирменный стиль и специфику страховой деятельности. Особое внимание уделено сдержанной цветовой гамме, которая вызывает доверие и подчеркивает надежность бренда.

Архитектура веб-представительства страхового агентства была разработана на основе модели «клиент-сервер», что обеспечивает четкое разделение логики работы клиента и сервера. Это решение позволяет создать масштабируемую, безопасную и удобную для поддержки систему. Клиентская часть отвечает за отображение интерфейса пользователя и взаимодействие с сервером через HTTP-запросы, используя современные технологии (HTML, CSS, JavaScript, Bootstrap). Серверная часть реализована с использованием Flask, что обеспечивает гибкость и производительность.

База данных спроектирована с учетом требований к функциональности и нормализации данных, что обеспечило эффективное функционирование системы. Применение принципов третьей нормальной формы позволило минимизировать дублирование данных и повысить целостность информации.

Интерфейс веб-представительства разработан с акцентом на удобство использования и адаптивность. Применение современных технологий (HTML, CSS, JavaScript, Bootstrap) позволило создать интуитивно понятный и привлекательный дизайн. Формы и элементы интерфейса позволяют пользователям легко выполнять ключевые действия: регистрацию, авторизацию, расчет стоимости полиса и оформление заказов.

Маршрутизация реализована с использованием Flask, что обеспечивает четкую организацию навигации между страницами. Абсолютные и относительные пути применяются в зависимости от структуры приложения и предпочтений разработчика. Это улучшает удобство использования и поддержку системы.

Тестирование Веб-приложения проведено на нескольких уровнях:

- модульное тестирование – проверка корректности работы отдельных функций и модулей;
- интеграционное тестирование – проверка взаимодействия между компонентами системы;
- системное тестирование – оценка работы приложения в целом.
- тестирование на отказоустойчивость – проведено 10 тысяч запросов, процент ошибок составил 2,95%, что подтверждает стабильность работы системы при высоких нагрузках;
- также было проведено тестирование безопасности с использованием ZAP 2.16.1, что позволило выявить и устранить потенциальные уязвимости.

На основе результатов тестирования были внесены следующие улучшения:

- устранен повторяющийся код (принцип DRY);
- оптимизированы запросы к базе данных, что снизило время отклика системы;
- добавлены индексы для повышения производительности запросов;
- улучшен пользовательский интерфейс для повышения удобства взаимодействия;
- реализована аутентификация и авторизация через Flask-Login;
- защита от CSRF-атак обеспечена через Flask-WTF.

Таким образом, в ходе реализации и тестирования веб-представительства страхового агентства были успешно решены все поставленные задачи. Созданное приложение является надежным, удобным и готовым к использованию в реальных условиях. Проведенные доработки значительно повысили производительность, безопасность и удобство использования системы, что делает ее конкурентоспособной на рынке страховых услуг.

Заключение

В рамках выполнения выпускной квалификационной работы была успешно разработана и реализована информационная система веб-представительства страхового агентства. Проект направлен на автоматизацию процессов взаимодействия между клиентами и страховым агентством, а также на упрощение управления данными для администраторов.

Основной целью работы было создание удобного и функционального веб-представительства, который предоставляет возможность пользователям ознакомиться с каталогом страховых продуктов, рассчитать стоимость полиса с помощью онлайн-калькулятора, оформить заявку на страхование, получить доступ к личному кабинету для просмотра истории заказов.

Для администраторов реализованы инструменты для управления списком страховых продуктов (добавление, редактирование, удаление), изменения статусов заказов, управления офисами компании (добавление, редактирование, удаление).

Все поставленные задачи были успешно выполнены, что подтверждается работоспособностью системы и проведенным тестированием.

Основные результаты работы – был проведен анализ существующих решений страховых компаний, выявлены их преимущества и недостатки. На основе анализа сформированы требования к системе.

Разработаны:

- логическая модель данных (диаграмма классов, вариантов использования, последовательности);
- физическая модель базы данных;
- архитектура приложения на основе паттерна MVC.

Реализация функционала. Используются современные технологии:

- Flask (Python) для создания серверной части;

- SQLAlchemy для работы с базой данных;
- HTML/CSS/JavaScript для клиентской части;
- Werkzeug для обработки файлов и безопасности;
- Nominatim API для отображения офисов на карте.

Проведено тестирование системы на производительность с использованием JMeter. Система показала стабильную работу при нагрузке до 200 одновременных пользователей.

Преимуществом разработанной системы являются:

- удобство для пользователей – интуитивно понятный интерфейс, возможность рассчитать стоимость полиса онлайн, личный кабинет для просмотра истории заказов;
- удобство для администраторов – простое управление продуктами, заказами и офисами через административную панель, возможность загрузки изображений для страховых продуктов;
- масштабируемость – система легко адаптируется под новые требования (например, добавление новых типов страховых продуктов или интеграция с платежными системами);
- безопасность – использование хеширования паролей для защиты данных пользователей, проверка прав доступа для предотвращения несанкционированных действий;

Несмотря на успешную реализацию проекта, есть направления для дальнейшего улучшения:

- интеграция с платежными системами – реализация возможности оплаты полисов онлайн;
- добавление мобильного приложения – создание мобильной версии сайта или нативного приложения для удобства пользователей;

- расширение функционала калькулятора – добавление дополнительных параметров для расчета стоимости полиса;
- интеграция с CRM-системой – автоматизация обработки заказов и взаимодействия с клиентами;
- оптимизация производительности – использование кэширования для ускорения работы системы при высокой нагрузке;

Разработанная система позволяет страховым агентствам:

- увеличить прозрачность и удобство взаимодействия с клиентами;
- снизить операционные затраты за счет автоматизации процессов;
- повысить конкурентоспособность на рынке страховых услуг.

Проект полностью соответствует поставленным целям и задачам.

Разработанное веб-представительство страхового агентства является удобным, безопасным и масштабируемым решением.

Реализация проекта способствует повышению эффективности работы страхового агентства и улучшению качества обслуживания клиентов.

Таким образом, выполненная работа представляет собой значимый вклад в развитие цифровых технологий в сфере страхования.

Список используемой литературы и используемых источников

1. Бабич, А. В. Введение в UML : учебное пособие / А. В. Бабич. – 4-е изд. – Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2022. – 198 с. – ISBN 978-5-4497-1637-8.
2. Баланов, А. Н. Бэкенд-разработка веб-приложений: архитектура, проектирование и управление проектами: учеб. пособие / А. Н. Баланов. – Санкт-Петербург [и др.] : Лань, 2024. – 311 с. – ISBN 978-5-507-48818-6.
3. Баланов, А. Н. Комплексная информационная безопасность : учеб. пособие / А. Н. Баланов. – Санкт-Петербург [и др.] : Лань, 2024. – 397 с. – ISBN 978-5-507-49250-3.
4. Белый, Е. М., Романова И. Б. Управление проектами : конспект лекций / Е. М. Белый, И. Б. Романова. – Москва : Ай Пи Ар Медиа, 2023. – 95 с. – ISBN 978-5-4497-1879-2.
5. Бух Г., Рамбо Дж., Якобсон И. Язык UML. Руководство пользователя / Г. Бух, Дж. Рамбо, И. Якобсон ; перевод Н. Мухина. – 3-е изд. – Москва : Академия АйТи, ДМК Пресс, 2022. – 494 с. – ISBN 978-5-89818-247-2.
6. ГОСТ 19.701-90 (ИСО 5807-85). Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения. – Дата введения 01.01.1992. – Изд. официальное. – Москва: Стандартформ, 2010. – 157 с.
7. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств. – Утвержден и введен в действие Приказом Росстандарта от 30.11.2010 № 631-ст. – Изд. официальное. – Москва: Стандартформ, 2011. – 106 с.
8. Грекул, В. И., Денищенко Г. Н., Коровкина Н. Л. Управление внедрением информационных систем : учеб. пособие / В. И. Грекул, Г. Н. Денищенко, Н. Л.

Коровкина. – 4-е изд. стер. электрон. – Москва : Интернет-Университет Информационных Технологий [и др.], 2025. – 277 с. – ISBN 978-5-4497-0910-3.

9. Гринберг М. Разработка веб-приложений с использованием Flask на языке Python / М. Гринберг. – ДМК-Пресс, 2016. – 272 с. – ISBN 978-5-97060-206-5.

10. Доусон М. Програмируем на Python / М. Доусон. – СПб.: Питер, 2019. – 416 с.

11. Документация Flask. Официальная документация Flask. URL: <https://flask.palletsprojects.com> (дата обращения: 29.04.2025).

12. Золотов С.Ю. Проектирование информационных систем : учебное пособие / С.Ю. Золотов. – Томск : Томский государственный университет систем управления и радиоэлектроники, Эль Контент, 2013. – 88 с. – ISBN 978-5-4332-0083-8.

13. Липаев, В. В. Программная инженерия сложных заказных программных продуктов : учеб. пособие / В. В. Липаев ; Институт системного программирования РАН. – Москва : МАКС Пресс, 2014. – 308 с. – ISBN 978-5-317-04750-4.

14. Машкин, А. В. Технология разработки программного обеспечения : учебное пособие / А. В. Машкин. – Вологда : ВоГУ, 2014. – 75 с. – ISBN 978-5-87851-526-9.

15. Муртазина, М. Ш. Управление проектами в сфере информационных технологий : учеб. пособие / М. Ш. Муртазина. – Новосибирск: Новосибирский государственный технический университет, 2022. – 61 с. – ISBN 978-5-7782-4618-8.

16. Программно-аппаратные средства защиты информации : учеб. пособие / С. А. Зырянов [и др.]. – Новосибирск : Новосибирский государственный технический университет, 2023. – 79 с. – ISBN 978-5-7782-4905-9.

17. Kim, G., Humble, J. Accelerate: Building and Scaling High Performing Technology Organizations– IT Revolution Press, 2023 (2nd ed). – ISBN: 978-1457191435.
18. Farley, D., Firth, M. Continuous Delivery Pipelines: Design, Implement, Measure, and Optimize Your Software Pipeline – O'Reilly Media, 2022. – ISBN: 979-8-5179-7955-1.
19. Beyer, B., Jones, C., Petoff, J., Murphy, N. Site Reliability Engineering: How Google Runs Production Systems – O'Reilly Media, 2022 (Updated Edition). – ISBN: 978-1663728586.
20. Zimmermann, C., Kliev, S., Hensley, A. Cybersecurity for Beginners – Independently published, 2023. – ISBN: 979-8868208157.
21. Python Software Foundation. Python Language Reference, version 3.12 Documentation / Официальная документация языка Python, версия 3.12. – URL: <https://docs.python.org/3.12/> (дата обращения: 03.03.2025).

Приложение А

Изображения интерфейса веб-представительства

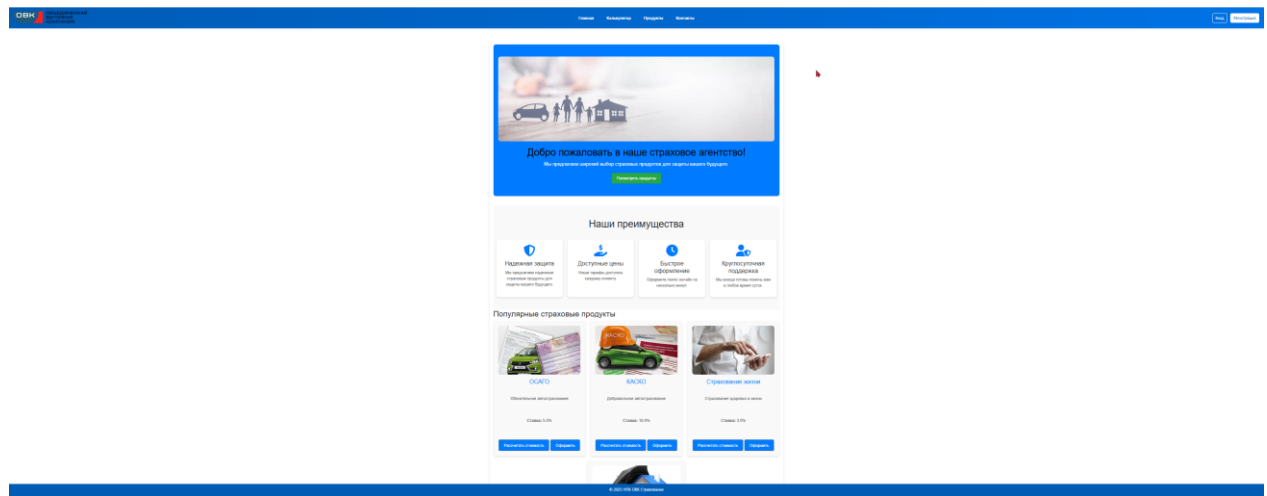


Рисунок А. 1 – Главная страница сайта

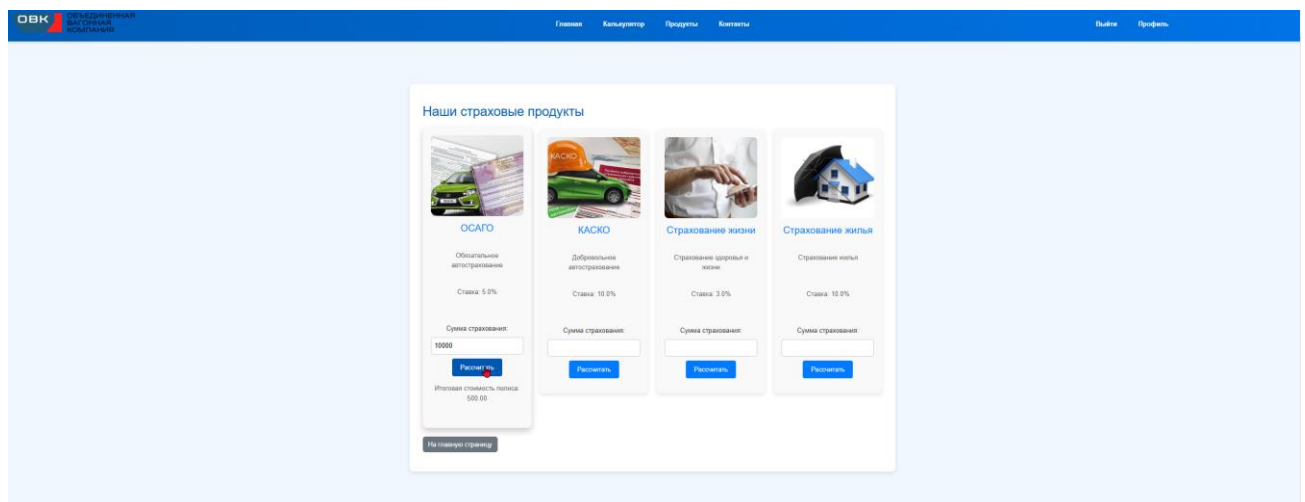


Рисунок А.2 – Каталог страховых продуктов (Products)

ОБК ОБЪЕДИНЕННАЯ
ВАГОННАЯ
КОМПАНИЯ

Главная Калькулятор Продукты Контакты Выход Профиль

Страховой калькулятор

Выберите страховой продукт и укажите сумму для расчета стоимости полиса.

Выберите страховой продукт:
ОСАГО (0.0%)

Сумма страхования (руб.):
20000

Рассчитать

Результат:
Стоимость полиса: 1000.00 руб.

Рисунок А.3 – Калькулятор стоимости полиса (Calculator)

ОБК ОБЪЕДИНЕННАЯ
ВАГОННАЯ
КОМПАНИЯ

Главная Калькулятор Продукты Контакты Выход Профиль

Личный кабинет пользователя: Вячеслав

Имя пользователя: Вячеслав
Email: qazex296@gmail.com

Список ваших страховых продуктов:

Название продукта	Сумма страхования	Итоговая стоимость	Статус	Дата создания
ОСАГО	10000.0 руб.	500.0 руб.	new	2025-05-01 18:33

[На главный экран](#)

Рисунок А.4 – Личный кабинет пользователя (Profile)

ОБК ОБЪЕДИНЕННАЯ
ВАГОННАЯ
КОМПАНИЯ

Главная Калькулятор Продукты Контакты Вход Регистрация

Регистрация

Имя пользователя

Адрес электронной почты

Пароль

Повторите пароль

Зарегистрироваться

Рисунок А.5 – Форма регистрации

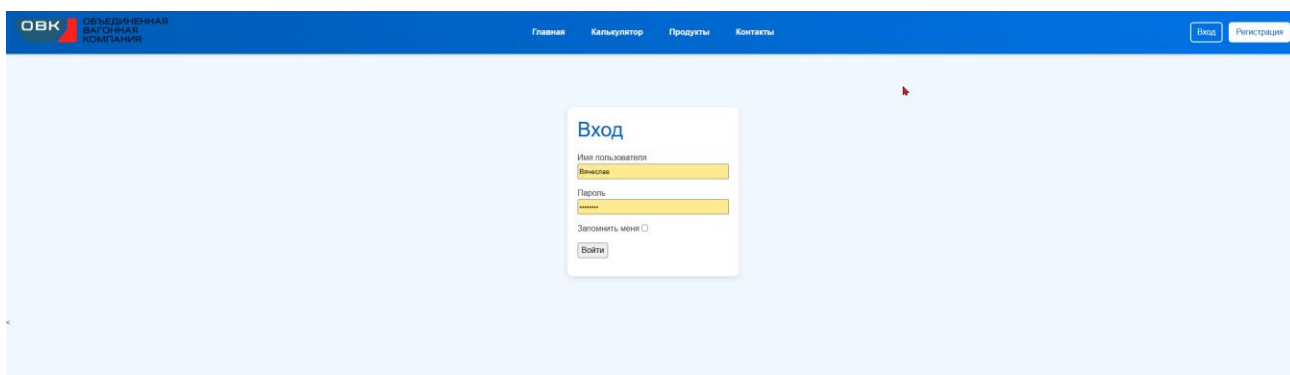


Рисунок А.6 – Форма авторизации

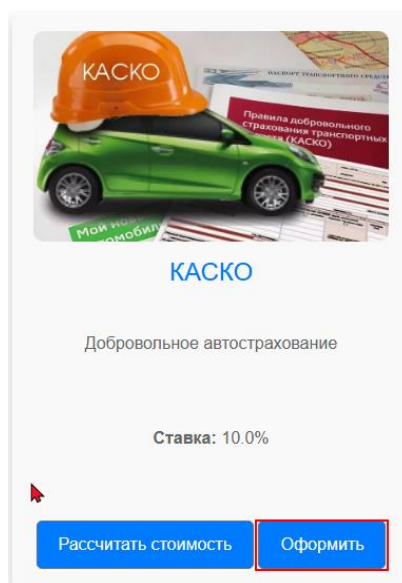


Рисунок А.7 – Выбор страхового продукта

Оформление заказа: КАСКО

Категория: Автострахование

Описание: Добровольное автострахование

Ставка страхования: 10.0%

Введите сумму страхования:

[Оформить](#) [На главную страницу](#)

Рисунок А.8 – Подтверждение заказа

Панель администратора

Страховые продукты

[Добавить продукт](#)

ID	Название	Описание	Цена (%)	Категория	Действия
1	ОСАГО	Обязательное автострахование	5.0%	Автострахование	Редактировать Удалить
2	КАСКО	Добровольное автострахование	10.0%	Автострахование	Редактировать Удалить
3	Страхование жизни	Страхование здоровья и жизни	3.0%	Страхование жизни	Редактировать Удалить

Рисунок А.9 – Таблица страховых продуктов в панели администратора

Добавление нового страхового продукта

Название:

Описание:

Цена (%):

Категория:

Изображение: Выберите файл Файл не выбран

[Добавить](#) [Вернуться в панель администратора](#)

Рисунок А.10 – Добавление/редактирование продукта

Офисы

Добавить офис

ID	Название	Адрес	Широта	Долгота	Действия
1	офис продаж	Россия, Москва, Арбат, 10	55.7514182	37.5962391	<button>Редактировать</button> <button>Удалить</button>

Рисунок А.11 – Таблица управления офисами в панели администратора

Заказы пользователей

ID	Пользователь	Продукт	Сумма страхования	Итоговая стоимость	Статус	Дата создания	Действия
1	123	ОСАГО	20000.0 руб.	1000.0 руб.	completed	2025-04-21 20:51	Завершен ▾ <button>Сохранить</button>
2	Вячеслав	КАСКО	10000.0 руб.	1000.0 руб.	new	2025-05-01 18:48	Новый ▾ <button>Сохранить</button>
3	Вячеслав	КАСКО	10000.0 руб.	1000.0 руб.	new	2025-05-01 18:51	Новый ▾ <button>Сохранить</button>

Рисунок А.12 – Управление заказами в панели администратора

Управление пользователями

ID	Имя пользователя	Email	Текущая роль	Изменить роль
1	admin	admin@example.com	admin	Администратор ▾ <button>Изменить</button>
2	123	okolinvyacheslav@yandex.ru	admin	Администратор ▾ <button>Изменить</button>
3	ZAP	zaproxy@example.com	user	Пользователь Администратор <button>Изменить</button>
4	Вячеслав	qazex298@gmail.com	user	Пользователь ▾ <button>Изменить</button>

Рисунок А.13 – Управление пользователями в панели администратора