АННОТАЦИЯ

Пояснительная записка к дипломному проекту содержит постановку и программу решения задачи «Интерактивная карта города Уфа». Программа написана на фреймворке Node.js, React, CSS. В качестве системы управления базы данных использовался MySQL,

Программа предназначена для работы в браузерах любой операционной систем.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Интерактивная карта — это современный инструмент визуализации данных, позволяющий отображать географическую информацию в удобной и наглядной форме. В условиях роста объёмов данных и необходимости их оперативного анализа, использование интерактивных карт становится важным элементом для оптимизации работы компаний и организаций. Такой подход позволяет быстрее обрабатывать данные, выявлять закономерности, отслеживать изменения в режиме реального времени и принимать обоснованные решения.

Для эффективного использования географических данных требуется решение, которое позволит визуализировать информацию в динамичном формате, обеспечивая интерактивность и доступность. Интерактивная карта предоставляет пользователям возможность фильтровать, группировать и анализировать данные, получая наглядное представление о территориальном распределении объектов или событий.

Целью данной работы является разработка интерактивной карты, которая будет включать функционал для визуализации и анализа данных, удобную навигацию, а также инструменты для фильтрации и сортировки информации. Внедрение такого решения позволит повысить эффективность работы с пространственными данными и улучшить процесс принятия решений.

1. Проектирование информационной системы
   1. Описание предметной области

Информационная система создаётся для автоматизации работы страховой фирмы. Страховая компания предлагает различные виды страхования, такие как автомобильное, здоровья и имущества. Основная цель страховой фирмы - обеспечить клиентам защиту от различных рисков и возмещение убытков в случае наступления страхового случая.

Страховая компания работает по нескольким основным принципам. Во -первых, клиенты могут выбирать и оформлять страховые полисы на различные виды страхования. Во-вторых, компания предлагает различные условия и тарифы для каждого вида страхования.

Пользователями информационной системы страховой фирмы являются:

* клиенты;
* агенты по продаже страховых полисов;
* администраторы.

Клиенты могут воспользоваться следующими функциями системы:

* просмотр информации о доступных видах страхования;
* оформление страхового полиса;
* выбор программ или условий;
* подача заявления на страховое возмещение.

Агенты по продаже страховых полисов имеют доступ к следующим функциям системы:

* просмотр информации о клиентах;
* оформление новых страховых полисов;
* обновление информации о существующих полисах;
* расчёт страховых взносов;
* обработка заявлений на страховое возмещение.

Администраторы имеют доступ к следующим функциям системы:

* управление списком агентов и их данных;
* управление списком доступных видов страхования;
* генерация отчётов о продажах, статистике и финансовых показателях.

Основные сущности в системе:

* клиенты;
* агенты по продаже страховых полисов;
* страховые полисы;
* виды страхования;
* заявления на страховое возмещение.

У клиента есть свои атрибуты:

* уникальный код;
* фамилия;
* имя;
* контактная информация.

У страхового полиса есть свои атрибуты:

* номер полиса;
* тип страхования;
* дата начала действия полиса;
* окончания действия полиса;
* страховая сумма.

У вида страхования есть свои атрибуты:

* название;
* описание;
* условия;
* тарифы.

При оформлении страхового полиса клиентом, информация о полисе отправляется агенту по продаже страховых полисов, в которой видно:

* информация о клиенте и его контактных данных;
* выбранный вид страхования и его условия;
* дата начала и окончания действия полиса.

Ограничения в информационной системе:

* для оформления страхового полиса клиент должен быть зарегистрирован в системе;

1.2 Описание входной информации

Входная информация служит для осуществления деятельности системы, в которую входят данные пользователя, данные о полисе. Информация о пользователе формируется на основе:

* фамилия клиента;
* имя клиента;
* отчество клиента;
* телефон;
* дата рождения;
* электронная почта.

Также входной информацией является данные о новом полисе, которые заносятся в базу данных в момент создания нового полиса администратором и характеризуются:

* название;
* цена;
* описание;
* категория.

1.3 Описание выходной информации

Выходная информация — информация, которая возникает в результате обработки человеком или устройством входной информации.

Основные требования к выходной информации сводятся к обеспечению пользователей сайта данными в удобной для них форме. После обработки всей входной информации на выходе получается документ, содержащий в себе данные о полисе клиента, также на почту отправляется документ.

Описание выходных документов представлено в таблице 1.3.1

Таблица 1.3.1 - Описание выходных документов

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование документа(шифр) | Периодичность выдачи документа | Кол-во экземпляров | Куда передаются |
| Полис страхования | При одобрении агентом заявки на получение полиса | 1 | Клиенту |

1.4 UML - диаграммы

UML – диаграмма — это схема, нарисованная с применением символов UML. Она может содержать множество элементов и соединений между ними.

Полное описание масштабного проекта может состоять из нескольких UML-диаграмм, связанных или не связанных между собой.

1.4.1 Диаграмма прецедентов

Диаграмма прецедентов (или диаграмма вариантов использования) в UML — это диаграмма, отражающая отношения между актёрами и прецедентами.

Она является составной частью модели прецедентов и позволяет описать систему на концептуальном уровне.

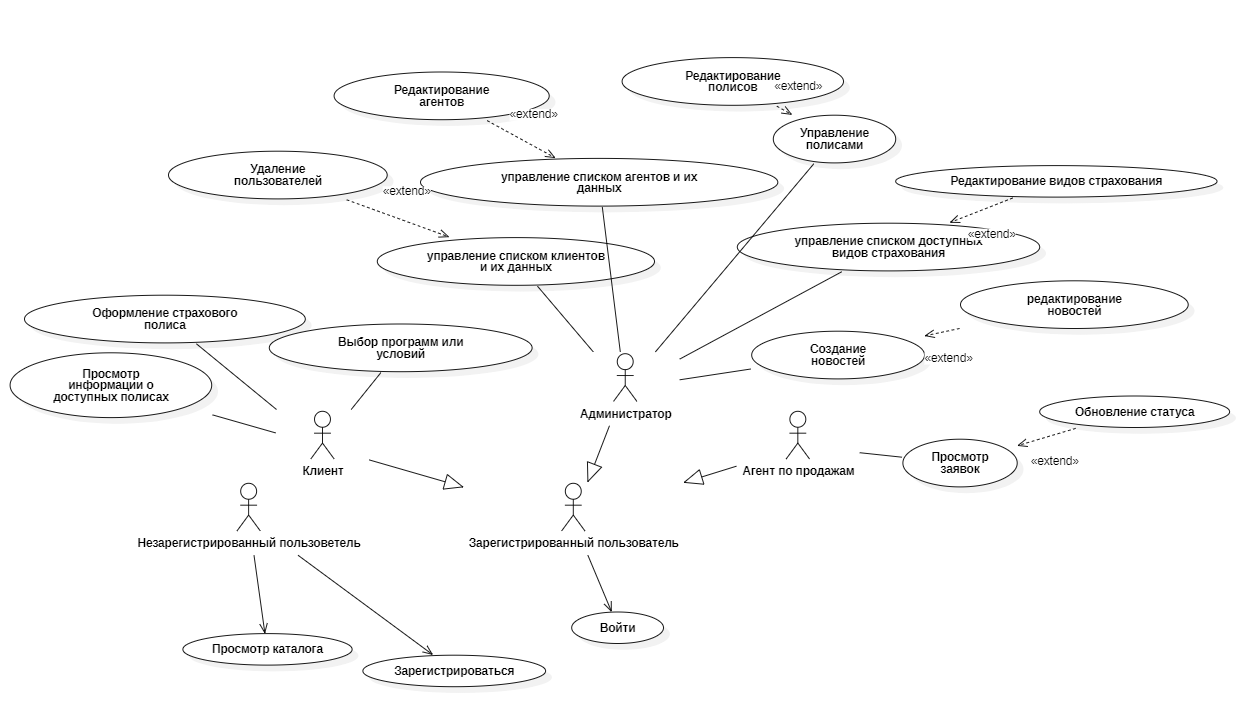


Рисунок 1.4.1.1 – Концептуальное моделирование

1.4.2 Диаграмма классов

Диаграмма классов ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA" \o "Английский язык) class diagram) — структурная [диаграмма](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_(UML)" \o "Диаграмма (UML)) языка моделирования [UML](https://ru.wikipedia.org/wiki/UML" \o "UML), демонстрирующая общую структуру иерархии [классов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)" \o "Класс (программирование)) системы, их коопераций, [атрибутов](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B5_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0" \o "Поле класса) (полей), [методов](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_(%D1%8F%D0%B7%D1%8B%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)" \o "Метод (языки программирования)), интерфейсов и взаимосвязей (отношений) между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования.

Диаграмма классов представлена на рисунке 1.4.2.1

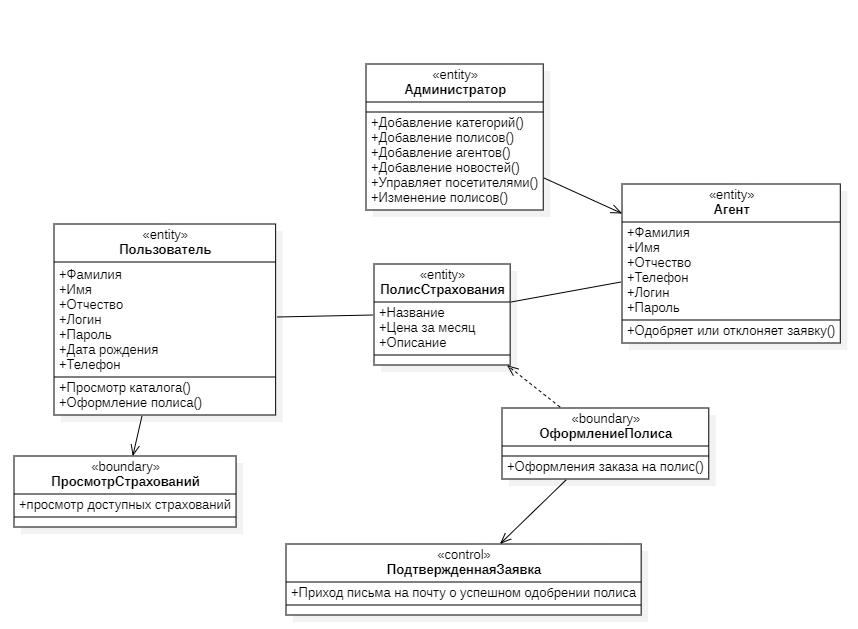


Рисунок 1.4.2.1 – Диаграмма классов

1.5 Концептуальное моделирование

Концептуальное моделирование – это процесс создания абстрактного представления о системе, которое фокусируется на том, что система должна делать, а не на том, как она это будет делать.

Концептуальное моделирование представлена на рисунке 1.5.1

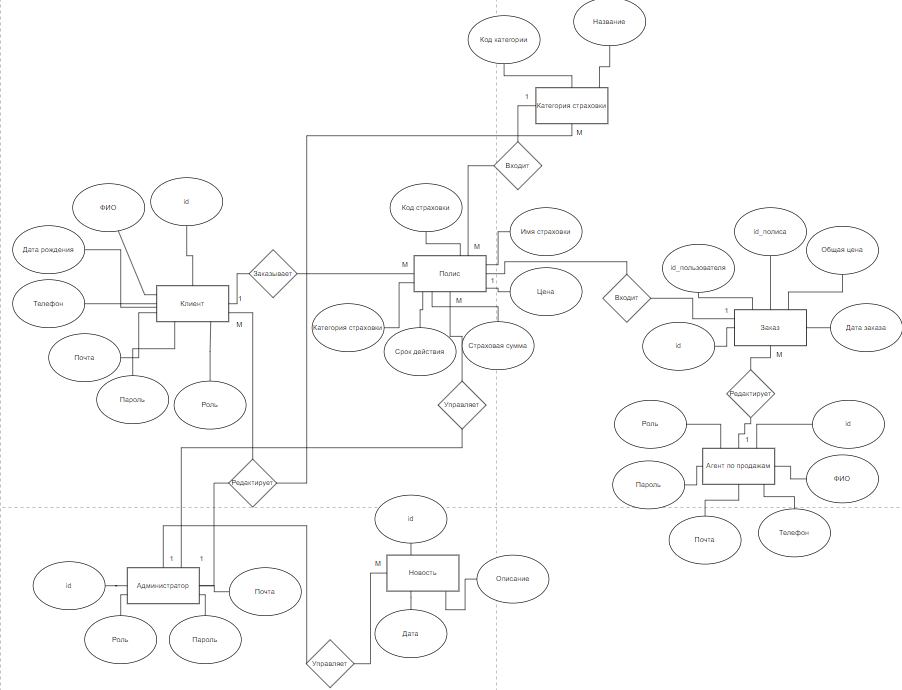


Рисунок 1.5.1 – ER-модель нотации Чена

1.6 Логическое моделирование

Логическое моделирование — это процедура проверки функционирования логической схемы с помощью компьютера.

Его основная цель — проверить функцию проектируемой логической схемы без её физической реализации, поскольку после изготовления схемы внести изменения в неё при современной технологии нелегко и недешево.

Логическое моделирование представлено на рисунке 1.6.1

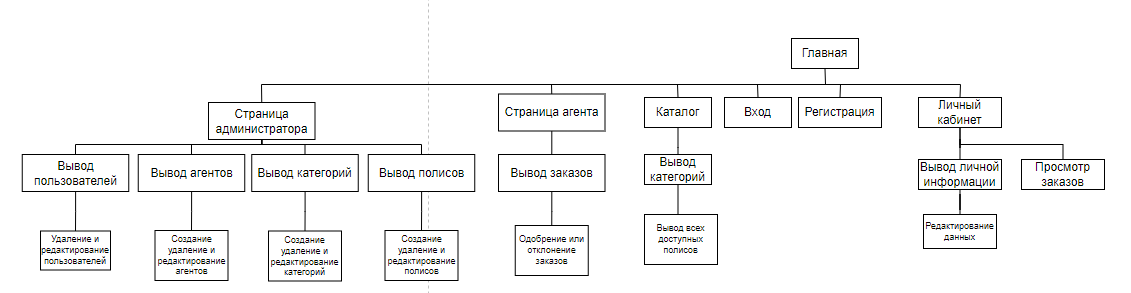


Рисунок 1.6.1 – Логическое моделирование

1.7 Описание структуры базы данных

Схема отношений базы данных представлена на рисунке 1.7.1

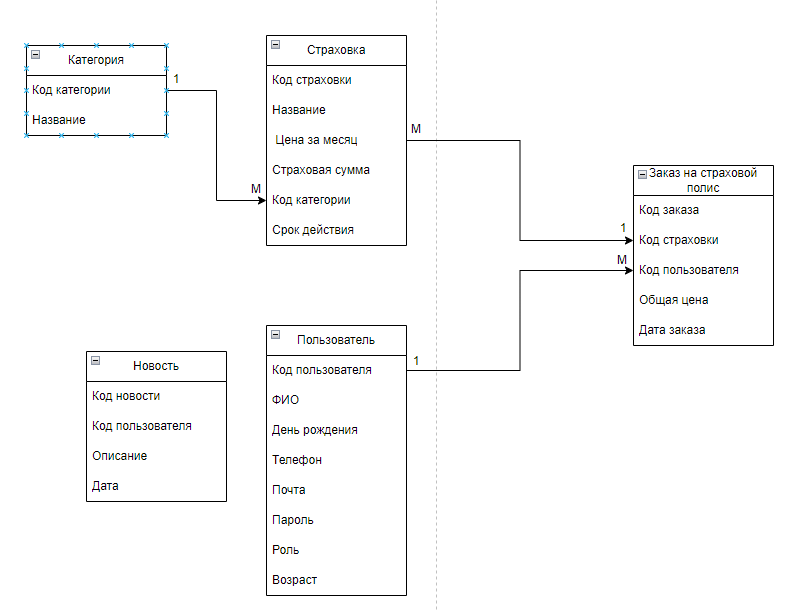


Рисунок 1.7.1 – Схема отношений базы данных

Структура таблицы users представлена в таблице 1.7.1

Таблица 1.7.1 - users (Пользователи)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| 1 | 2 | 3 | 4 |
| Код пользователя | id | INT | Первичный код |

Продолжение таблицы 1.7.1.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| ФИО пользователя | FIO | VARCHAR (150) |  |
| Дата рождения | date\_of\_birthday | DATE | Необязательное поле |
| Телефон | phone | VARCHAR (11) |  |
| Логин | email | VARCHAR (50) |  |
| Пароль | password | VARCHAR (50) |  |
| Роль | role | ENUM('user', 'admin', 'agent') | Принимает значение: user, admin, agent |
| Возраст | age | INT | Необязательное поле |

Структура таблицы news представлена в таблице 1.7.2

Таблица 1.7.2 - news (Новости)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Код новости | id\_news | INT | Первичный ключ |
| Описание | descr | VARCHAR (250) |  |
| Дата | date\_of\_publication | DATE |  |
| Изображение | image | VARCHAR (50) |  |

Структура таблицы order представлена в таблице 1.7.3

Таблица 1.7.3 - order (Заказы)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| 1 | 2 | 3 | 4 |

Продолжение таблицы 1.7.3

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Код заказа | order\_id | INT | Первичный код |
| Код страховки | insurance\_id | INT | Вторичный ключ |
| Код пользователя | id\_user | INT | Вторичный ключ |
| Цена | total\_price | VARCHAR (11) |  |
| Дата заказа | date\_of\_order | VARCHAR (50) |  |
| Срок действия | validity\_period | VARCHAR (50) |  |
| Статус | status | ENUM('Одобрена', 'Отклонена', 'Новая') | Принимает значение: Одобрена, Отклонена, Новая |

Структура таблицы insurance представлена в таблице 1.7.4

Таблица 1.7.4 – insurance (Страховка)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Код страховки | insurance\_id | INT | Первичный код |
| Имя страховки | insurance\_name | VARCHAR (150) |  |
| Цена | price | DATE |  |
| Код категории | category\_id | VARCHAR (50) | Вторичный ключ |

Структура таблицы category представлена в таблице 1.7.5

Таблица 1.7.4 – category (Категория)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Код категории | category\_id | INT | Первичный код |
| Имя категории | category\_name | VARCHAR (50) |  |