2 СТРУКТУРНО-ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ

2.1 Структурное проектирование

Структура проекта представляет собой трёхуровневую архитектуру. Трёхуровневая архитектура – архитектура клиент-серверных приложений, которая разбивает приложения на три уровня:

- представление (клиент);
- логика (сервер);
- данные (БД).

Каждый уровень приложения должен быть независимым и закрытым, то есть не должен показывать зависимости, которые относятся к реализации данного уровня. Все слои должны быть соединены между собой, значит несоединённые слои не должны иметь возможности взаимодействовать.

Уровень представления предоставляет пользовательский интерфейс, управляет взаимодействием с пользователем и не должен содержать никакой бизнес-логики или доступа к данным.

Уровень логики содержит в себе набор правил для обработки информации, может работать с несколькими пользователями и не должен содержать в себе уровень доступа и данных.

Уровень данных содержит в себе физическое хранилище данных, обрабатывает запросы к базе данных и не должен содержать в себе к уровню представления и логики.

Использование трёхуровневой архитектуры в веб-приложениях подразумевает то, что на уровне представления отрисовывается статический или динамический контент, на уровне логики существует сервер приложения для динамической обработки данных, а на уровне данных присутствует база данных, которая хранит в себе данные и имеет возможность управления этими данными.

Структура трёхуровневой архитектуры представлена на рисунке 2.1

Клиентское приложение представляет собой одностраничное приложение (Single page application, SPA). SPA – тип веб-приложений, особенность которых заключается в том, что данное приложение использует единственный HTML-документ в качестве оболочки для всех веб-страниц и организует динамическую подгрузку HTML, CSS, по запросу клиента через AJAX-запросы.

Огромным плюсом таких приложений является то, что они не отрисовывают заново целую страницу с данными, а лишь обновляют данные на странице, что увеличивает скорость работы и ответа сервера таких приложений. Также SPA могут

работать на огромном количестве устройств, будь то стационарный компьютер или смартфон, или даже телевизор.

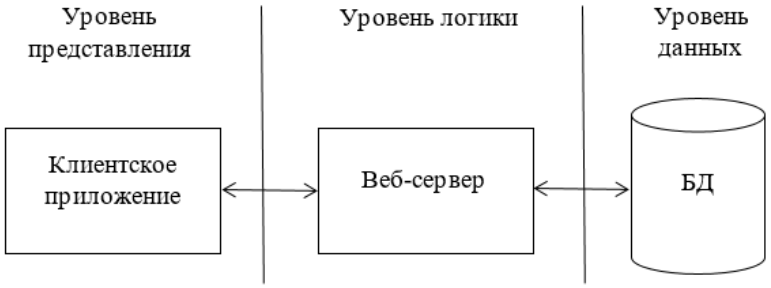

Рисунок 2.1 – Трёхуровневая архитектура приложения

При построении SPA придерживаются нескольких принципов:

− наличие клиентской навигации, хождения по которой сохраняются в истории навигации для удобного перехода по ранее сохранённым ссылкам;

− так как SPA – это приложение одной страницы, то все скрипты и стили должны быть определены в этой единственной странице;

− SPA загружает все скрипты, требующиеся для старта и инициализации веб-приложения;

− подгрузка модулей осуществляется постепенно по требованию клиента.

Веб-сервер будет построен по архитектуре REST. Задача веб-сервера – по запросу клиента возвращать корректные данные посредством запросов в базу данных.

Доступ к клиентскому приложению будут иметь любые пользователи, однако функциональность будет отличаться в зависимости от роли пользователя в системе.

Доступ к веб-серверу будет имеет только разработчик для внесения правок в функционал приложения. Сервер будет обращаться к базе данных, в которой хранится вся информация о всех сущностях приложения.

Таким образом, получена структура, согласно которой клиент делает запрос на веб-сервер для получения данных, веб-сервер делает запрос в базу данных, база данных возвращает данные веб-серверу, который обрабатывает их и передаёт клиенту.

Схема ресурсов разработанного программного обеспечения представлена на чертеже БрГТУ.141144 - 07 90 00.

2.2 Функциональное проектирование

В ходе работы единственными входными данными для приложения будут являться реквизиты пользователей (имя пользователя и пароль для входа, а также

дополнительные данные при регистрации), данные, требующиеся для идентификации пользователя в системе.

Для хранения всех данных приложения будет использоваться база данных. Выделим основные сущности и связи между ними - построим концептуальную модель базы данных.

В ходе проектирования базы данных были выделены следующие сущности:

- пользователи;
- маршруты;
- транспортные средства;
- остановки;
- поездки пользователей;
- избранные маршруты пользователей.

При проектировании концептуальной модели у сущностей были выделены атрибуты, которые определены в таблице 2.1.

Таблица 2.1 – Атрибуты сущностей КМ БД

Название атрибута	Является ли ключом	Обязателен	Тип данных	Описание
1	2	3	4	5
Пользователь				
id	PK	+	ObjectId	Первичный ключ
cardNumber	-	+	String	Номер счёта пользователя
loginName	-	+	String	Имя пользователя в системе
password	-	+	String	Пароль пользователя
role	-	+	String	Роль пользователя в системе
active	-	+	Boolean	Статус пользователя
bill	-	+	Number	Баланс пользователя
Маршрут				
id	PK	+	ObjectId	Первичный ключ
number	-	+	String	Номер маршрута
vehicleType	-	+	String	Тип транспортного средства
stops	-	+	[String]	Остановки в прямом направлении
stopsReverse	-	+	[String]	Остановки в обратном направлении
Транспортное средство				
id	PK	+	ObjectId	Первичный ключ
number	-	+	String	Номер транспортного средства
route	-	+	String	Ссылка на маршрут

Продолжение таблицы 2.1

1	2	3	4	5
Остановки				
id	PK	+	ObjectId	Первичный ключ
name	-	+	String	Название остановки
coordinates	-	-	[String]	Координаты остановки (широта, долгота)
Поездка				
id	PK	+	ObjectId	Первичный ключ
user	-	+	String	Ссылка на пользователя
vehicle	-	+	String	Ссылка на транспортное средство
from	-	+	String	Начальная точка
to	-	+	String	Конечная точка
payment	-	+	Number	Оплата
date	-	+	Date	Время поездки
Избранный маршрут				
id	PK	+	ObjectId	Первичный ключ
user	-	+	String	Ссылка на пользователя
from	-	+	String	Начальная точка
to	-	+	String	Конечная точка

На основе концептуальной модели базы данных построим логическую модель базы данных. Построение схемы проводилось с помощью Microsoft Access. Реализация логической модели представлена на рисунке 2.2.

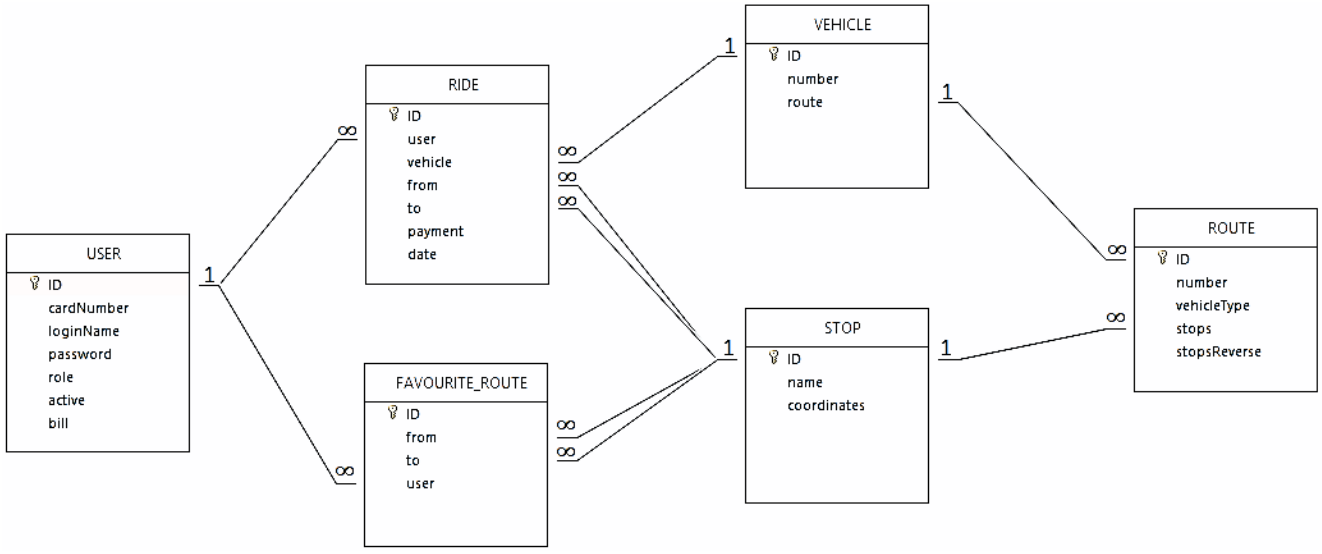

Рисунок 2.2 – Модель «сущность-связь» базы данных

Ограничение передачи данных клиенту осуществляется на сервере. При авторизации для каждого клиента будет генерироваться JSON Web Token (JWT). JWT – специальный ключ, состоящий из 3 частей, который позволяет обезопасить передачу данных между двумя источниками. В основном, JWT генерируется по алгоритму HS256.

Как уже упоминалось, ключ состоит из трех частей, разделённых точками:

− Заголовок – представляет собой JSON-объект, состоящий, в основном, из двух полей: тип алгоритма для шифрования и тип токена. После описания, данный JSON-объект шифруется в Base64 формат и образовывает первую часть токена.

− Данные – JSON-объект, который, как правило, состоит из данных о пользователе (например, ID или имя пользователя) и сервисной информации. Как и заголовок, данный объект шифруется в формат Base64 и формирует вторую часть токена.

− Подпись – третья часть токена. Для её создания требуется, чтобы были зашифрованы Заголовок и Данные, выбран алгоритм шифрования и был предоставлен секретный ключ – некоторая строка, которая повлияет на шифровку и дешифровку токена.

JWT генерируется на сервере и отправляется клиенту при авторизации. На клиенте этот ключ хранится в локальном хранилище или в виде куки. При создании запросов к серверу от клиента в заголовок протокола HTTP добавляется поле, которое содержит значение этого ключа. Когда запрос приходит на сервер, он расшифровывает данный ключ и определяет, отдавать какие-либо данные текущему пользователю или нет.

Чтобы получить данные с API, требуется сделать запрос по определённому конечному пути. В таблице 2.3 приведено описание путей.

Таблица 2.2 – Описание API

Метод	Путь	Значение параметров	Описание
1	2	3	4
/auth			
POST	/signin	-	Авторизация
POST	/signup	-	Регистрация
/user			
GET	/all	-	Получить всех пользователей
GET	/me	-	Получить информацию о пользователе
GET	/favorites	-	Получить избранные маршруты пользователя
PUT	/put-money	-	Пополнить баланс пользователя

Продолжение таблицы 2.2

1	2	3	4
/ ride			
POST	/	-	Добавить поездку
GET	/user-all	-	Получить все поездки пользователей
/ route			
POST	/	-	Добавить новый маршрут
PUT		-	Изменить маршрут
DELETE	/:id	id – ID маршрута	Удалить маршрут
GET	/all	-	Получить все маршруты
GET	/:id	id – ID маршрута	Получить маршрут
/stop			
GET	/all	-	Получить все остановки
POST	/	-	Добавить остановку
PUT		-	Изменить остановку
DELETE		-	Удалить остановку
/vehicle			
POST	/	-	Добавить новое ТС
PUT		-	Изменить ТС
DELETE		-	Удалить ТС
GET	/all	-	Получить все ТС
GET	/:id	id – ID маршрута	Получить ТС

На клиенте благодаря использованию Vuex все данные хранятся в одном контейнере, называемом «хранилище». Все данные в «хранилище» доступны только для чтения, поэтому, чтобы его изменить, надо будет провести следующие действия:

− клиент жмёт на кнопку, тем самым делая запрос на сервер;

− находится «действие», которое описывает поведение данной кнопки, и вызывается «диспетчер», который обрабатывает текущее «действие»;

− когда вызывается действие, вместе с ним отрабатывает «мутация», в которой описано, как и какие данные нужно изменить в «хранилище» при ответе с сервера.

Согласно ТЗ в разрабатываемом проекте должны осуществляться оплата проезда, определение местоположения пользователя и отображение выбранного маршрута на карте.

Есть разные варианты, как настроить на сайте оплату: напрямую через банки, с помощью платежных систем, платежных шлюзов, операторов сотовой связи или агрегаторов платежных сервисов. Важным фактором при выборе наравне с безопасностью проведения платежей и размером комиссии должен стать комфорт осуществления процесса оплаты для покупателя, а также простота и скорость подключения сервиса.

Принято выделять пять основных способов оплаты на сайте:

− оплата банковскими картами, онлайн-операции по которым обрабатываются процессинговыми центрами;

− оплата с использованием электронных кошельков от операторов электронных денежных средств (ЭДС);

− оплата с помощью мобильных платежей с баланса телефонного счета посредством операторов сотовой связи;

− оплата посредством сервисов интернет-банкинга – технология проведения платежей e-invoicing, которая позволяет выставлять электронные счета покупателям товаров и услуг;

− оплата наличными через кассы и терминалы – оплата посредством стационарных или мобильных платежных терминалов.

Для оплаты проезда пользователю необходимо будет предварительно пополнить баланс. Пополнение баланса будет осуществляться путем введения данных о пользовательской карте: номер карты, CSV-код и срок годности. Затем необходимо будет указать сумму и нажать на кнопку «подтвердить».

Для принятия платежей на стороне сайта необходимо будет подключить специальный плагин одной из банковских систем. Для этого необходимо предварительно зарегистрироваться и привязать карту, на которую будут осуществляться переводы денежных средств. В данном проекте будет использоваться плагин от поставщика «stripe.com», поскольку он является одним из первых в своем роде, а также имеет множество положительных отзывов.

Для работы с картой можно использовать готовые решения (API) от таких производителей, как Yandex или Google. В данном проекте будут использованы «Яндекс.Карты», поскольку они наиболее информативны для нашей местности.

Определение местоположения будет осуществляться с помощью встроенных аппаратно-программных средств мобильных телефонов – систем позиционирования. Чем больше систем позиционирования поддерживает устройство, тем выше будет точность местоположения.

Спутниковая система навигации – система, предназначенная для определения местоположения наземных, водных и воздушных объектов. Спутниковые системы навигации также позволяют получить скорости и направления движения приёмника сигнала, кроме того могут использоваться для получения точного времени. Такие системы состоят из космического оборудования и наземного сегмента (систем управления). В настоящее время только две спутниковые системы обеспечивают полное и бесперебойное покрытие земного шара – GPS и ГЛОНАСС.

Принцип работы спутниковых систем навигации основан на измерении расстояния от антенны на объекте, координаты которого необходимо получить, до спутников, положение которых известно с большой точностью. Таблица

положений всех спутников называется альманахом, которым должен располагать любой спутниковый приёмник до начала измерений. Обычно приёмник сохраняет альманах в памяти со времени последнего выключения и, если он не устарел мгновенно использует его. Каждый спутник передаёт в своём сигнале весь альманах. Таким образом, зная расстояния до нескольких спутников системы, с помощью обычных геометрических построений на основе альманаха можно вычислить положение объекта в пространстве [3].

Для определения местоположения пользователя необходимо подтвердить разрешение на получение его GPS-координат. После получения географических координат они будут переданы в виде долготы и широты навигационной системе «Яндекс.Карты», которая отобразит местоположение пользователя с точностью до 6 - 8 метров.

2.3 Разработка структуры пользовательского интерфейса

Пользовательский интерфейс представлен в виде одной страницы, в которой в зависимости от пути и данных динамически изменяется содержимое. В зависимости от роли в системе, пользователь имеет доступ к определённым данным. Права доступа определяется на уровне серверной части. Права доступа к задачам определены в таблице 2.3.

При первом входе на сайт пользователь будет неавторизированным, а также ему будет показываться главная страница сайта. Если пользователь авторизируется и не выйдет из системы, то при перезагрузке сайта или при повторном его посещении пользователь всё еще будет авторизирован благодаря тому, что его данные будут храниться в локальном хранилище браузера.

Право «пользователь» будет назначаться всем пользователям, зарегистрировавшимся в системе при помощи клиентского приложения. Право «администратор» получит первый пользователь зарегистрировавшийся в системе, также он сможет назначать других администраторов.

Неавторизованный пользователь сможет только авторизоваться или зарегистрироваться. Для авторизации пользователю необходимо ввести логин и пароль, а для регистрации дополнительно номер карты.

Для построения маршрута объявления пользователь должен будет перейти на вкладку «Маршруты», которая находится в панели навигации.

Пользователь должен быть авторизирован в системе. На странице создания маршрута (смотри рисунок 1.10) пользователю потребуется заполнить данные

о маршруте, начальной точке и конечной точке. После ввода данных пользователь сможет посмотреть построенный маршрут на карте, а также транспорт, который проходит через эти точки. Также на данной вкладке пользователю будут доступны избранные маршруты и возможность к их добавлению.

На вкладке «Профиль» пользователь сможет увидеть состояние своего счёта, пополнить его, посмотреть историю поездок, а также оплатить поездку. Для оплаты поездки пользователю необходимо будет указать тип ТС, после чего система найдет и предложит пользователю на выбор транспортные средства, которые находятся поблизости с ним в радиусе 50 метров. Внутри ТС будет указан уникальный номер ТС, который необходимо ввести в приложение. Затем необходимо будет заполнить информацию о начальной и конечной точке поездки и нажать на кнопку «оплатить».

Таблица 2.3 – Права доступа к системе

Операция	Без прав	Пользователь	Администратор
Регистрация	+		
Авторизация	+		
Просмотр карты	-	+	+
Определение местоположения	-	+	+
Построение маршрута	-	+	+
Добавление в избранное	-	+	+
Просмотр истории поездок	-	+	+
Пополнение счёта	-	+	+
Оплата поездок	-	+	+
Генерация QR-кода электронного билета	-	+	+
Просмотр маршрутов	-	+	+
Добавление маршрутов	-	-	+
Редактирование маршрутов	-	-	+
Удаление маршрутов	-	-	+
Просмотр ТС	-	+	+
Добавление ТС	-	-	+
Редактирование ТС	-	-	+
Удаление ТС	-	-	+
Просмотр остановок	-	+	+
Добавление остановок	-	-	+
Редактирование остановок	-	-	+
Удаление остановок	-	-	+
Просмотр всех пользователей	-	-	+
Назначение ролей	-	-	+

При клике на поездку система сгенерирует QR-код электронного билета, который можно будет предоставить в качестве оплаты за проезд.

Одним из преимуществ системы будет то, что пользователь может оплачивать поездки даже после того, когда закончатся деньги, однако с ограничением в 2.5 рубля. При достижении ограничения пользователь будет автоматически заблокирован и разблокируется только после следующего пополнения счёта.

Для того, чтобы пополнить счёт, пользователю необходимо нажать на кнопку «пополнить счёт» из своего профиля, ввести данные кредитной карты: номер, срок годности, код CVV, сумму на которую он хочет пополнить баланс и нажать на кнопку «подтвердить».

На вкладке «карта» пользователь сможет определить своё положение, найти объекты на карте, построить маршрут, увидеть, где находится транспорт.

Администратору будет доступен весь функционал приложения, однако его главная роль – это управление системой. Ему будут доступны функции управления (добавление, редактирование и удаление) для всех элементов системы: маршруты, транспорт, остановки и пользователи.

На главной странице администратора будут доступны все маршруты с детальной информацией: описание, количество ТС на маршруте, тип ТС, количество перевезенных пассажиров и прибыль. Будет возможность фильтрации по типу ТС и по временному диапазону (для статистической информации). На каждой строке маршрута будут кнопки для редактирования и удаления.

На странице управления транспортом администратору будут доступны все зарегистрированные ТС, а также возможность добавления нового ТС. Для добавления нового ТС будет необходимо указать его номер (должен быть уникальным), тип ТС и также выбрать маршрут, по которому он будет проходить и нажать на кнопку «зарегистрировать тс». На каждой строке ТС будут кнопки для редактирования и удаления.

На странице управления остановками администратору будут доступны все зарегистрированные остановки, а также возможность добавления новой остановки. Для добавления новой остановки будет необходимо ввести ее название и координаты. На каждой строке остановки будут кнопки для редактирования и удаления.

На странице управления пользователями администратору будет доступна вся информация о пользователях: номере карт, имена, роли, статусы и счета. Здесь также будет доступно общее количество пользователей, и сколько из них активны/заблокированы. Администратор сможет изменять роли пользователей.

Схема данных разработанного программного обеспечения представлена на чертеже БрГТУ.141144 - 07 91 00.