

РАЗРАБОТКА ИНТЕРАКТИВНОГО СЕРВИСА ГРУЗОПЕРЕВОЗОК

В статье описаны инструментальные средства разработки и указано их непосредственное применение для создания интерактивного сервиса грузоперевозок, используемого для создания заказа, отслеживания перемещения груза на карте в реальном времени, установления конкретных временных рамок для доставки груза и многого другого.

В настоящее время современный мир и его технологии развиваются крайне активно и быстро, так же, как развивается рынок услуг и его потребности. С развитием современных технологий, появилось понимание, что многие вещи, особенно в бизнесе, можно и нужно оптимизировать, например, логистику – один из ключевых моментов любого рода деятельности [1].

Значение автомобильного транспорта трудно переоценить, этот вид транспорта становится всё более и более популярным. Большая потребность и востребованность в грузоперевозках объясняется тем, что этот вид транспорта не только способен перевозить все виды грузов, но также обладает уникальной среди остальных видов транспорта особенностью – доставка груза от двери до двери.

Для создания заказа грузоперевозки (рисунок 1), необходим начальный адрес (т. е. координаты) и пункт назначения. Так же необходимо точно установить какое-либо определенное количество перемещаемого груза. В транспортной модели предполагается, что стоимость перевозки по какому-либо маршруту прямо пропорциональна объему груза, перевозимому по этому маршруту. Непосредственно построение маршрута происходит с помощью стороннего посредника, сервиса Routific, которому передается объект с координатами остановок.

Хранение данных в силу большого числа преимуществ происходит с помощью NoSQL – базы данных Cloud Firestore, выпущенной в 2017 году, которая была разработана специально для хранения и синхронизации данных веб-приложений на глобальном уровне. Для разработки Front части приложения используется фреймворк Angular 2, управление состоянием данных и интерфейса происходит с помощью Redux. Серверная часть приложения написана на C#, весь сервис развертывается на платформе облачных приложений следующего поколения Microsoft Azure Service Fabric [2].

После регистрации, где пользователь указывает свои контактные данные и свой адрес, который будет по умолчанию первоначальным адресом отправления последующих грузов (этот адрес можно изменять как в профиле пользователя, так и при создании любого заказа), появляется возможность создания заказа. Прежде всего, на первоначальном этапе создания заказа, указывается стартовый адрес, так же необходимо выбрать тип заказа (distribution или collection), после того как адрес и тип будут выбраны, появится возможность перехода на следующий этап оформления заказа. При переходе на следующий этап (по нажатии кнопки continue), на сервер отправится POST запрос, который будет содержать в себе данные о заказе, с которыми он инициализирован.

На сервере происходит обработка этого запроса и создание заказа в базе данных с соответствующими значениями. Так же на сервере происходит присваивание уникального зашифрованного id каждому заказу и каждой остановке заказа, для дальнейшего их распознавания. Ответ сервера будет содержать в себе данные созданного заказа с присвоенным уникальным id.

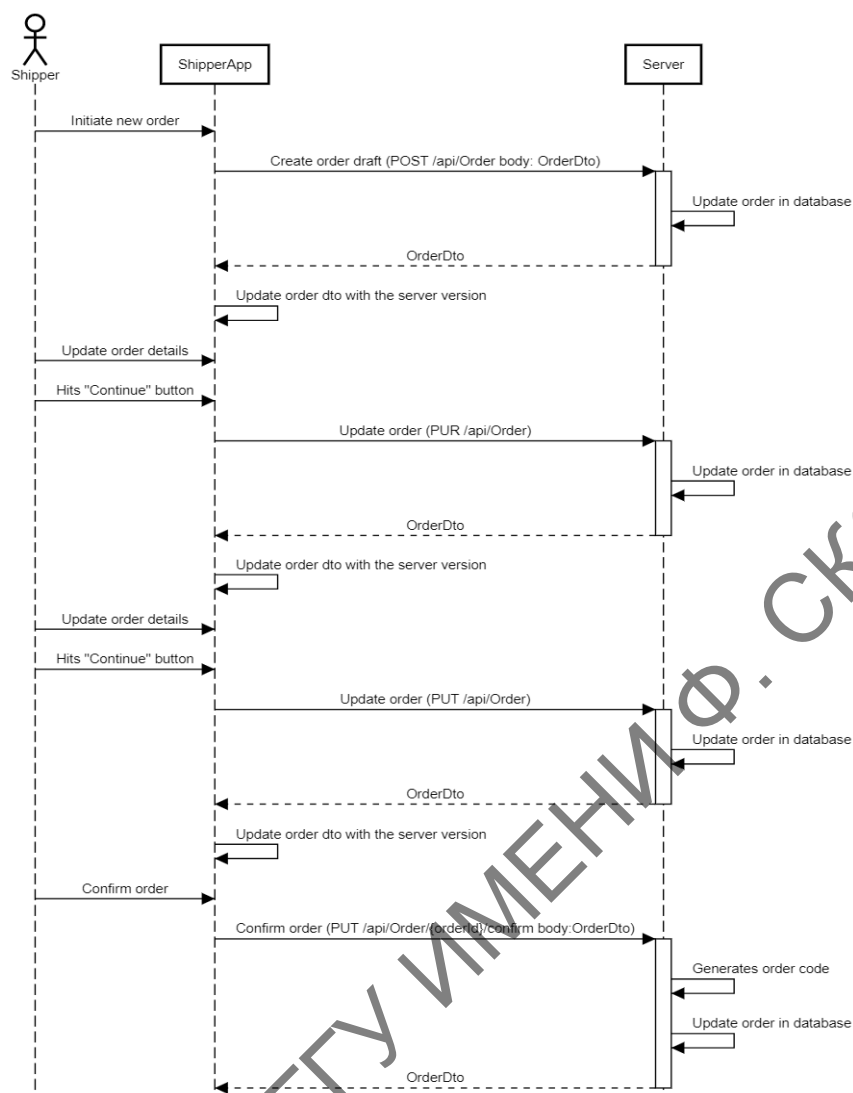


Рисунок 1 – Схема создания заказа

На следующем этапе необходимо ввести детали заказа, такие как контактное имя, телефон, электронный адрес и т.д. После того как будут введены необходимые данные, при переходе на следующий этап так же будет отправлен PUT запрос на сервер, содержащий в себе всю информацию о заказе (ту, что уже была введена на прошлом этапе, а так и же и ту, что была введена на текущем этапе), и по этому запросу необходимые данные конкретного заказа будут обновлены в базе данных. Распознавание заказа происходит с помощью уникального поля id. После того, как вся информация о первоначальной (отправной) остановке будет введена, отправлена и обработана сервером, необходимо ввести данные о следующей остановке, куда требуется отвезти груз (в случае типа заказа distribution) или забрать еще один груз (в случае типа заказа collection).

Алгоритм создания последующих остановок идентичен, только в зависимости от типа заказа, будут и отличаться типы последующих остановок – будут ли они остановками типа delivery (куда доставлять груз) или остановками типа pickup (где груз забирать).

Следующий этап – это этап оплаты (рисунок 2). Оплата происходит банковской картой с помощью стороннего сервиса для оплаты, таким образом, база данных не хранит личных данных о картах пользователей, что предотвращает возможность утечки персональных данных.

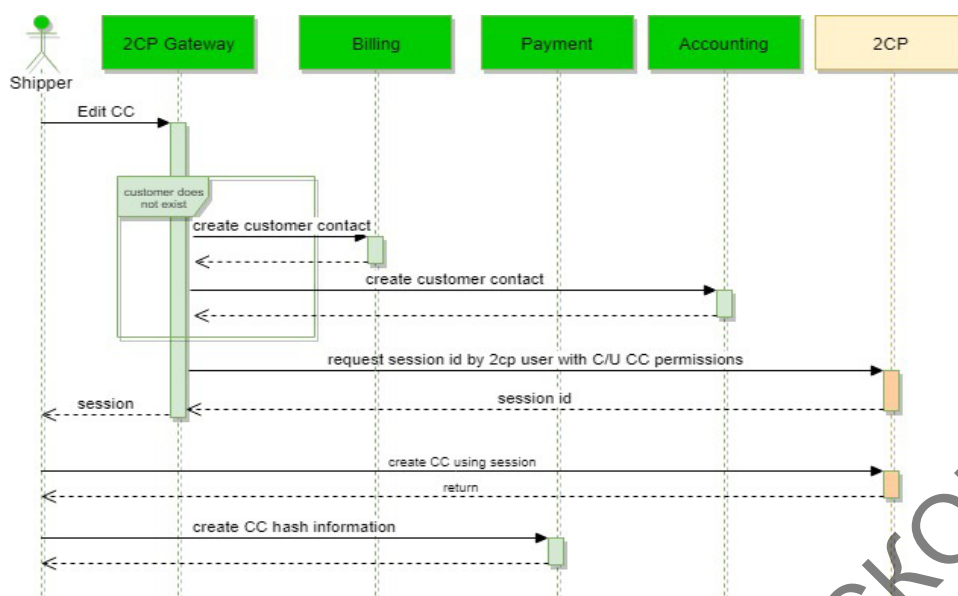


Рисунок 2 – Схема процесса оплаты заказа

Большинство запросов посылаются на сервер и обрабатываются асинхронно, для увеличения производительности и скорости работы, где это возможно, несколько запросов оптимизируются в один, но это не исключает того, что в таком типе приложений особенно важно постоянно обновлять данные.

Так же в сервис была добавлена поддержка уведомлений, приходящих в браузер или на мобильные девайсы в виде всплывающих уведомлений. Для поддержки данной возможности, потребовалось перевести сайт сервиса с протокола HTTP, на протокол с безопасностью HTTPS, что никак не сказалось на продуктивной работе и других возможностях сервиса.

Для процесса оплаты используется 2C Processor (2CP). 2CP использует токенизацию (ключи доступа), позволяя совместным клиентам хранить случайное число, связанное с данными конфиденциальной кредитной карты клиента, а не фактический номер кредитной карты.

Так же в сервис была добавлена возможность отслеживания перемещения заказа (машины водителя, который его перевозит) в режиме реального времени (рисунок 3). Происходит это следующим образом – в заполняемой форме о данных заказа необходимо указать email и поставить галочку «Отслеживать заказ». После того, как все данные о заказе были верно введены, был выбран желаемый водитель для перевозки груза (или выбрана опция – любой свободный водитель на подходящем под параметры груза автомобиле) и совершена оплата заказа, на указанный email-адрес придет письмо со ссылкой на страницу отслеживания заказа, на которой будет отображена карта с текущими позициями водителя (зеленый маркер на карте) и позицией груза (цвет маркера зависит от типа остановки), адрес груза на карте тот, что был введен во время заполнения данных заказа.

В правой части экрана находится непосредственно информация о текущем статусе заказа (заказ создан, водитель принял заказ, груз подобран водителем, груз в пути, доставлено), информация о водителе, время создания заказа и время доставки груза, информация о клиенте, сообщения о событиях, происходящих с грузом и их время, информация о самом грузе (количество, вес, размер), который необходимо забрать или доставить на текущую остановку. При наличии каких-либо проблем с определенной единицей груза – сообщения о проблеме отображаются напротив данного объекта в раскрывающемся окне с информацией о текущей остановке.

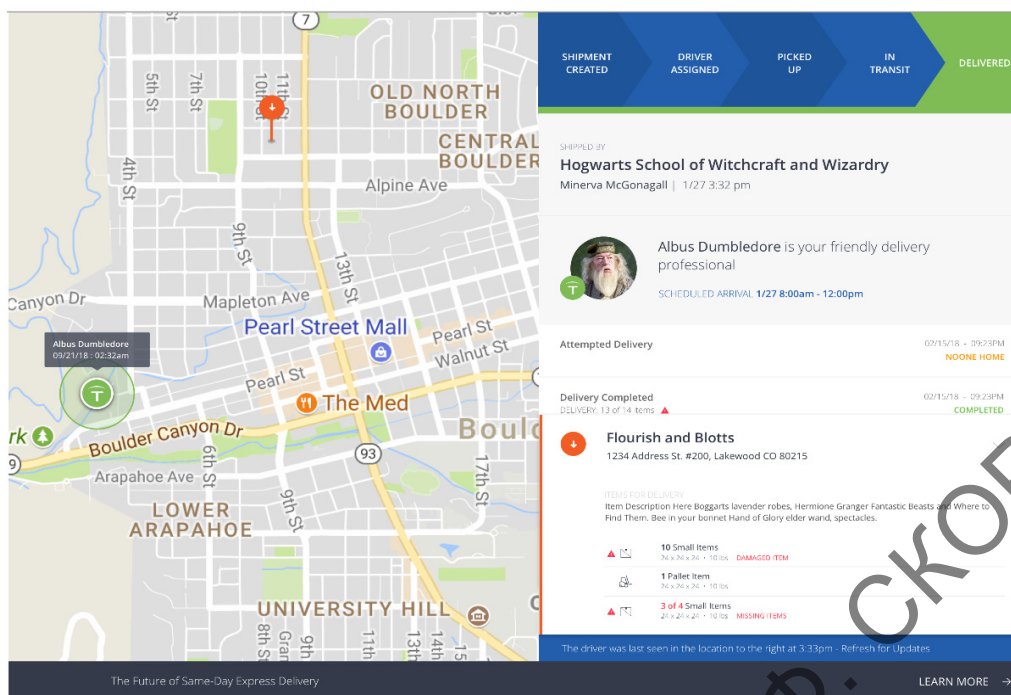


Рисунок 3 – Страница отслеживания заказа

Хранение данных в силу большого числа преимуществ происходит с помощью NoSQL – базы данных Cloud Firestore, которая была разработана специально для хранения и синхронизации данных веб-приложений на глобальном уровне. Для разработки Front части приложения используется фреймворк Angular 2. Серверная часть приложения написана на .Net, весь сервис разворачивается на платформе облачных приложений следующего поколения Microsoft Azure Service Fabric.

Оптимизация скорости работы и повышение надежности сервиса являлось и является одним из приоритетных условий разработки данного сервиса, т. к. он используется на многомиллионном рынке потенциальных клиентов.

Литература

1 Сукач, Е. И. Технология проектного моделирования транспортных систем региона, функционирующих в условиях случайных воздействий / Е. И. Сукач // Известия Гомельского государственного университета имени Ф. Скорины. – 2008. – № 5 (50). – С. 112–117.

2 Официальная документация Microsoft Azure Service Fabric [Электронный ресурс]. – Режим доступа: <https://azure.microsoft.com>. – Дата доступа: 1.03.2018.

УДК 530.182, 535-4, 535.012.2, 535.016, 535.518, 537.862, 537.872.32

Е. Д. Головин

МУЛЬТИПОЛЬНЫЕ МОМЕНТЫ ЗАРЯДОВ, РАСПОЛОЖЕННЫХ НА ОДНОЙ ПРЯМОЙ

В статье изложены расчеты потенциалов зарядов, расположенных на одной прямой. Показано, что специальным выбором величин зарядов и их конфигурации можно добиться, чтобы потенциал системы зарядов совпадал с потенциалом мультиполя, образованного зарядами, расположенными на плоскости или в пространстве.