**Архитектура и работа с приложением**

1. **Сущности в приложении**
   1. **Пользователь.**

- Пользователь создается на стороне сервера.

- С клиента производится только логин под конкретным пользователем. По

- У пользователя есть свойство “role”, которое может принимать значения {“driver”, “admin}.

- Водитель является основным пользователем клиентской части и действующим лицом.

- Если в кллиентском приложении произведен логинв аккаунт с ролью “admin”, в приложении для пользователя открывается доп. функционал.

- Пользователи не могут быть удалены, только помечаются как удаленные. Потому что к ним привязываются Смены, Визиты и прием товаров

- Логин производится по номеру телефона (определяется клиентом автоматически) и ПИН-коду.

- Свойство “phone\_number” используется для логина в аккаунт (по требованию заказчика). Поскольку пользователи устройств могут меняться, но не могут быть удалены, может возникнуть ситуация, когда с БД будет несколько записей пользователей с одинаковым телефоном. Для этого при управлении пользователями необходима валидация, что может быть только 1 неудаленный пользователь для каждого телефонного номера.

* 1. **Сайт**

- Магазин или любое другое место, откуда водители забирают товар

- Регистрируются на стороне сервера

- В клиентском приложении отображаются имя и уникальный цифровой код. Возможно адрес (но это неточно)

- имеет координаты для валидации визитов

- UUID – уникальная строка по которой генерируется QR-код магазина – используется для валидации клиентом. Если QR-код будет распространаться от сервера, то заменить его на двоичный массив данных самого кода, который будет валидировать клиент

- Сайты не могут быть удалены, только помечаются удаленными. Потому что к ним привязаны визиты и, опосредованно, прием товара.

* 1. **Категории товаров**

Это то, что принимает водитель на сайте.

- из значимых параметров пока только имя.

- не могут быть удалены или изменены, только помечены удаленными. Потому что к ним привязаны приемы товара

* 1. **Цены**

- Коллекция цен с указанием даты установки и привязкой к категории товара

- эти сущности не удаляются

- актуальными считаются самые новые записи для каждой категории товара

- цена определяется двумя целочисленными параметрами: значением и масштабом, который определяет сколько младших десятичных разрядов из значения находятся в дробной части. Например: {value=42575, scale=2} => 425.70, {value=4257, scale=1} => 425.7

* 1. **Смена / Shift**

Основная сущность, которая генерится на сервере в процессе работы и определяет связь водителя и работы на конкретную дату.

- Смена имеет свой жизненный цикл, который описывается свойством “state”.Оно имеет следующие значения: “new” – смена доступна для исполнения водителем. В этом состоянии ее можно редактировать (добавлять и удалять визиты); “accepted” – водитель принял смену к исполнению. В этом состоянии смену редактировать нельзя; “closed” – смена закрыта водителем либо после исполнения всех визитов либо вручную. Редактировать в этом состоянии нельзя; “cancelled” – это свойство проставляет сервер. Значит что смена удалена. В этом состоянии водитель не может принять ее к исполнению и она не возвращается ему в АПИ. Перейти в это состояние можно только из состояния “new” или из состояния “accepted” в крайнем случае.

- допустимые переходы смены между состояниями

**new**

**accepted**

**closed**

**cancelled**

client: start

client: close

webadmin: cancel

webadmin: cancel

client: reject

(optional)

- при создании смены на сервере необходимо проверять чтобы для каждого водителя на 1 день было не больше одной смены в состоянии “new”.

- Если для водителя сегодня уже есть смена в состоянии “accepted”, новую смену создать нельзя (но можно отменить текущую и создать новую).

- Для одного водителя на один день может быть несколько неактивных смен типа “cancelled” / “closed” и только 1 активная смена типа “new” / ”accepted”

- В смене хранятся метаданные по визитам для ускорения выдачи: общее колич. визитов, количество выполненных визитов, количество неудачных визитов.

* 1. **Визит / Visit**

Основная сущность с которой работает водитель. В рамках визита он принимает товар и редактирует его состояние.

- Визит определяет песещение конкретного сайта “site\_id” в рамках конкретной Смены “shift\_id” (соответственно устанавливается одгнозначная связь Сайт-Водитель-Дата).

- “CONSTRAINT shift\_site\_unique UNIQUE (shift\_id, site\_id)”

- Визит имеет порядковый номер в маршруте “order” и, опционально, желательнае время посещения сайта “suggest\_time”

- Визиты могут быть созданы/удалены/отредактированы только пока родительская смена находится в состоянии “new” !!!!!

- Визит имеет жизненный цикл, который определяется свойством “state” со следующими значениями: “new” – водитель еще там не был; “in\_progress” – водитель приехал и начал прием товара; “completed” – водитель удачно принял товар и закрыл визит; “failed” – водитель не смог принять товар.

- Состояние визита изменяет только клиентское приложение, но сервер обязан проверять валидность переходов

**new**

**in\_progres**

**completed**

**failed**

- свойство “start\_time” проставляется для визита сервером в момент перехода “new” -> “in\_progress”

- свойство ”complete\_time” проставляется сервером в момент “in\_progress” -> “completed”. При переходе в “failed” это свойство не заполняется.

- свойства “customer\_name”, “customer\_signature” заполняются клиентским приложением в рамках завершения визита с приемом товара.

* 1. **Маршрут/Route**

Виртуальная сущность не представленная отдельно в БД. Означает отсортированный по порядку набор визитов для Смены. Может фигурировать в АПИ

* 1. **Принятые товары /Collectedgoogs**

Сущности создаются клиентским приложением в рамках приема товара

- ID композитный определяет привязку к визиту “visit\_id” и категории товара “goodcategory\_id”

- также есть обязательная привязка к теблице цен “price\_id”, которая определяет цену по которой сегодня был принят этот товар

- Обязательное “weight” – вес принятого товара в килограммах, округленный до целого

- При создании этих сущностей в БД, сервер обязательно должен валидировать, что указанная цена действительно соответствует указанной категории товара (по таблице Prices) и является актуальной на текущую дату (самой новой).

- Создание списка принятыз товаров для визита производится одновременно с переводом состоянис визита “in\_progress” -> “completed” ТРАНЗАКЦИОННО !!!

- Если визит находится не в состоянии “in\_progress”, сощности Collectedgoods для него НЕ МОГУТ БЫТЬ СОЗДАНЫ !

* 1. **Заметка (комментарий) / Note**

Создается водителем в рамках исполнения смены. К заметке могут быть подвязаны фотографии. Заметка может быть создана только в рамках смены (например в пути), либо привязана также и к конкретному визиту в этой смене.

- имеет обязательную привязку к водителю, который являктся ее автором

- свойство дата-время создания – заполняется сервером

- имеет обязательную привязку к смене.

- для создания Заметки Смена должна быть в состоянии “accepted”

- имеет опционоальную привязку к визиту в рамках смены.

- при создании Заметки с одновременной привязкой к смене и визиту, тервер должан валидировать: что указанный визит действительно находится в указанной смене, Смена в состоянии “accepted”, Визит в состоянии “in\_progress”.

- Загрузка фотографий и создание Заметки производится в одном Multipart запросе транзакционно!

* 1. **NoteAttechment**

Фотографии к заметкам. Создаются при создании Заметки в рамках одной АПИ

1. **Client Application workflow**
   1. **Login**

Пользователь вводит на странице логина только свой ПИН-код. Номер телефона подставляется автоматически

* 1. Запуск главного экрана и поиск задач на сегодня
  2. Если задачи не найданы, показывается соответствующее сообщение. Водитель может перезапустить поиск, может завершить работу с приложением. Можно запустить приложение снова – поиск будет выполнено при старте.
  3. Если смена для этого водителя на сегодня найдена, ему показываются метаданные (кольчество запланированных визитов; возможно, длина маршрута :lol:) и большая красная кнопка “Yes, I’m ready to start the shift”.
  4. Нажав большую красную кнопку водитель получает маншрут этого шифта в виде списка визитов. Сам шифт переводится с состояние “accepted”
     1. Возможно наличие также кнопки “Reject”, но это неточно. По ее нажатию маршрут сразу закрывается без исполнения. Возможно, с автоматическим созданием комментария
  5. Водитель едет по маршруту. При приближении к сайту соответствующий визит, если он еще не исполнен, выделяется цветом.
  6. Водитель может в любой момент запостить Заметку на маршруте с главного экрана (для этого открывается отдельный экран редактора). Заметки и визиты отображаются на главном экране в одном списке вперемежку с сортировкой по времени.
  7. Водитель может в любой момент на маршруте с главного экрана выбрать опцию меню “close Shift”.
  8. После приезда на сайт водитель сканирует код (сканнер запускается с главного экрана нажатием на большую красную кнопку). Нужный визит выбирается автоматически по коду
  9. После удачного сканирования и валидации кода и выбора визита, происходит валидация его сайта по геолокации.
  10. После успешной валидации в 2.10 на сервер постится о начале визита и, в случае удачи, открывается форма заполнения данных для этого визита
  11. На форме показывается информация о сайте (валидируется водителем), информация о водителе (валидируется сайтом), форма для заполнения принятого товара.
      1. Если на этом етапе закрыть приложение, оно снова откроется на форме заполнения данных. Выйти с нее можно либо завершив прием товара либо отменив визит
      2. После этого повторно вернуться к этому визиту уже нельзя
  12. Находясь на форме заполнения данных водитель может написать заметку к визиту.
  13. Водитель может отменить визит со статусом “failed” в случае невозможности принять товар
  14. Если водитель принял товар, он предлагает сотруднику Сайта поставить подпись на форме
  15. После этого он постит все данные на сервер и визит завершается транзакционно.
  16. В случае удачи 2.16 печатается чек.
  17. Окно приема товара закрывается и водитель попадает обратно на главный экран на котором визит отмечен как выполненный
  18. При нажатии на выполненный визит откроются его детали с таблицей принятого товара, указанием имени приемщика, его подписью, фактически временем визита, именем и фотографией водителя, принявшего товар.
  19. Продолжаем с п. 2.6
  20. После исполнения всех визитов, клиентское приложение автоматически закрывает смену и выводит поздравление водителю
  21. Гемификация по результатам прохождения моршрута? (:LOL:)

1. **Замечания по логике работы приложения**
   1. **Получение данных о товарах и ценах**

Предложение – список товаров и актуальные цены загружаются в момент начала исполнения водителем маршрута и фиксируются на всю смену.

Альтернатива – обновлять список товаров и цены в начале исполнения каждого визита (это немного сложнее)

* 1. **Порядок исполнения визитов**

Визиты могут исполняться не в установленном порядке. В момент начала исполнения визита сервер проставляет для него время начала, которое будет определять реальную очередность прохождения маршрута. Для начатого визита запускается таймер, которой показывает водителю сколько он находится на сайте.

* 1. **Переход с выполняемого визита на главный экран.**

По умолчанию, после начала визита, приложение остается на экране заполнения данных о нем до завершения визита (или его отмены).

Но, возможно, это поведение может быть изменено (может будет локальная/удаленная настройка) и можно будет вернуться с незавершенного заполнения данных на главный экран. В таком случае, визит “in\_progress” будет отмечен в списке соответствующим образом с идущим таймером. Вернуться к заполнению данных можно будет нажатием на этот визит. Никакие другие действия на главном экране будут не доступны пока имеется незавершенный визит.

* 1. **Заполнение данных о визите**

При заполнении данных визита на экране должна показываться следующая информация:

- фотография и имя водителя для вализации заказчиком

- название и код магазина

- возможно, адрес магазина

- форма для заполнения товаров с указанием названия товара, текущей цены и принятого веса

- расчет итоговой стоимости

- кнопка для перехода к редактору заметок и отображение текущей написанной заметки и прикрепленных фотографий. К визиту может быть написана только 1 заметка (ограничение клиентского приложения для упрощения UI)

При выходе из недозаполненного визита введенные данные сохраняются и редактирование можно продолжить при следующем заходе. При этом визит продолжает считаться начатым.

При завершении визита появляется форма для ввода подписи. После того как форма подписана водитель нажимает кнопку «Завершить». Данные отсылаются на сервер, визит закрывается и распечатывается чек.

* 1. **Заметки привязанные к визитам**
     1. Для упрощения и ускорения разработки предлагаем разрешить только 1 заметку к визиту и вводить ее можно только во время заполнения данных визита – отправляется на сервер при закрытии визита вместе с данными о товаре.
     2. Дополнение. Можно добавить возможность написания заметок к уже завершенным визитам (если водитель что-то забыл указать изначально), но это дополнительная работа
  2. Ываывавапва
  3. вапва