Сессия 1

Основные задачи сессии 1:

* создание базы данных;
* создание центрального сервера;
* создание приложения АЗС:
* просмотр остатков топлива из БД
* создание/редактирование информации АЗС
* загрузка информации АЗС из БД через ЦС
* разработка модуля “Заправочная колонка”;
* разработка модуля “Камера”

**Создание базы данных**

Создайте базу данных, используя предпочтительную платформу (MySQL / Microsoft SQL Server), на сервере баз данных, который вам предоставлен. Используйте предоставленные в ресурсах к сессии данные для импорта. Цель базы данных - не только хранение информации, но и формирование отчетов по основным бизнес-процессам АЗС. Вам необходимо будет это учесть при проектировании вашего решения.

Работа с базой данных будет выполняться на протяжении всего проекта.  Определение основных сущностей, атрибутов и отношений будет выполняться вами во время 4 сессий. Проверка базы данных будет выполнена по итогам работы над всем проектом.

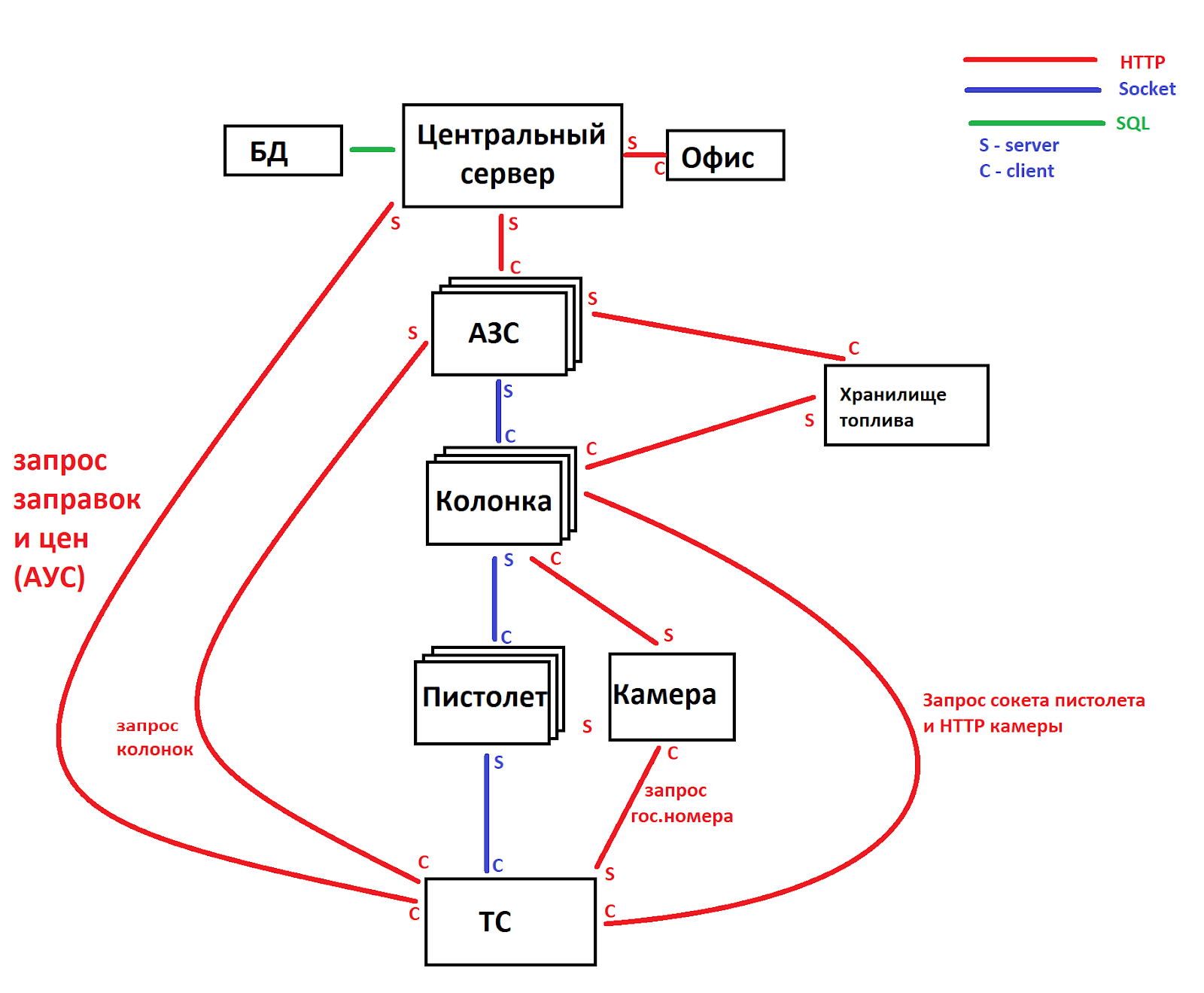
Минимальные требования к БД: импортированные данные,3 НФ, тестовые данные (5 записей) в каждой таблице, если нет данных для импорта.

Создайте таблицы основных сущностей, атрибуты, отношения и необходимые ограничения. После создания базы данных требуется импортировать предоставленные данные из папки «Ресурсы». Возможно, вам понадобится отформатировать данные, прежде чем загрузить их в таблицы, которые вы только что создали. В любом случае созданные таблицы должны содержать начальные тестовые данные.

При организации хранения данных вам необходимо учесть запрет на полное удаление данных, реализовав возможность отправки данных в архив. Кроме того, необходимо учесть, что данные о заказе не могут быть отправлены в архив, если не выполнена хотя-бы одна услуга в заказе.

Разработанная вами база данных должна быть в 3 НФ.

В последующих сессиях возможно вам понадобится добавить какие-либо сущности в ходе работы над проектом.

**Общая схема проекта**  
  
Обратите внимание - каждый модуль в указанной архитектуре - отдельное приложение. ваша задача - реализовать необходимые модули и обеспечить передачу данных согласно схеме с использованием http и websocket. Так же можно использовать формат передаваемых  данных -  JSON, csv и т.д.

**Модуль “Центральный сервер (ЦС)”**

Разработайте  приложение “Центральный сервер”

Центральный сервер является HTTP сервером, к которому подключаются все заправки (АЗС) в городе Санкт-Петербург и модуль “Офис”.

Только ЦС имеет подключение к базе данных.

Реализация данного модуля предполагает подход - Test Driven Development. Разработайте модульные тесты на каждый запрос, реализовав проверку на все возможные входные и выходные данные.

После реализации модульных тестов вам необходимо оформить документацию по модулю, описав структуру и методы запросов (см. пример в ресурсах).

**End-points:**

<http://127.0.0.1:8080/stations?fuel=FUELTYPE>   FUELTYPE может принимать значения 92, 95, 98, DT

Возвращает список станций АЗС, на которых поддерживается данный вид топлива.

Информация о каждой АЗС должна содержать:  
 - Географический адрес АЗС  
 - ID заправки

                - Цена на выбранный вид топлива

<http://127.0.0.1:8080/setStation>                  POST-запрос со следующими данными:  
 - ID заправки  
 - Географический адрес  
 - Поддерживаемое топливо (массив из объектов, содержащих типы топлива, цены на них и остаток в хранилище)

[http://127.0.0.1:8080/getStationInfo?i](http://127.0.0.1:8080/getStationInfo?id)d=STATION\_ID

STATION\_ID - идентификатор АЗС  
 Возвращает информацию, хранимую в базе о данной АЗС:  
 - Географический адрес

*http://127.0.0.1:8080/report?type= ???* - реализуйте дополнительные методы в ходе решения задания следующих сессий

**Модуль “АЗС”**  
 Задача модуля - управление АЗС.

Разработайте оконное приложение с минимальным графическим интерфейсом - только необходимые элементы для ввода и вывода данных.

Модуль “АЗС” является HTTP клиентом для Центрального Сервера.

При запуске приложения пользователь должен иметь возможность ввести STATION\_ID АЗС и нажать кнопку “Загрузить данные”.

По нажатию данной кнопки на Центральный сервер отправляется запрос [http://127.0.0.1:8080/getStationInfo?i](http://127.0.0.1:8080/getStationInfo?id)d=STATION\_ID

После получения ответа открывается окно “Управление АЗС №STATION\_ID”  
Окно должно содержать следующие редактируемые поля:  
- Географический адрес АЗС  
- Цена литра АИ-92 в рублях  
- Остаток АИ-92 в хранилище в литрах  
- Цена литра АИ-95 в рублях  
- Остаток АИ-95 в хранилище в литрах  
- Цена литра АИ-98 в рублях  
- Остаток АИ-98 в хранилище в литрах  
- Цена литра ДТ в рублях  
- Остаток ДТ в хранилище в литрах

При нажатии кнопки “Сохранить изменения” вышеуказанные данные отправляются на Центральный сервер POST-запросом <http://127.0.0.1:8080/setStation>Если информация по данной АЗС ранее сохранялась, она должна автоматически подгружаться в поля из БД. Если же АЗС с таким номером еще нет в базе, то при сохранении изменений в базе должна быть создана новая запись.

**Важно!!! STATION\_ID может принимать значение от 1 до 99.**

Реализуйте оконное приложение “Камера”.

Цель приложения - распознавание гос.номеров автомобилей с камер видеонаблюдения, установленных на АЗС.

Данное приложение - первое приложение, с которым клиент может работать на АЗС. Приложение будет установлено на терминале. Работа с приложением возможна только после звукового сигнала о нераспознанном гос. номере.

Когда транспортное средство останавливается у ТРК (топливо раздаточной колонки), с камер видео-наблюдения формируется фото ТС, которое видео-сервер отправляет в ЦОД: фотографию и распознанный по ней номер гос. регистрации в текстовом формате. Эти данные должны сохраняться в базе данных и инициализировать работу колонки.

При распознавании гос.номера сохраняются в базе данных с автоматическим присвоением номера электронной накопительной карты. Если же при распознавании гос номер в базе данных уже есть, то на модуль заправочной колонки отправляется сообщение (статус) о возможности заправки с помощью накопительной карты, причем клиенту не нужно вводить номер карты, он автоматически будет добавлен в процессе работы заправочной колонки.

Кроме того, у клиента должна быть возможность ручного внесения гос номера, если номер по фото не распознан.

Реализуйте в модуле “Камера” звуковое оповещение клиента о том, что гос номер не распознан, работа с накопительной картой невозможна. После звукового оповещения на терминале работа с интерфейсом становится доступной, и клиент может ввести свой гос номер, и загрузить фото, выбрав из 3, предложенных системой. Далее внесенный номер проверяется в базе данных, и если номер найден, то на модуль заправочной колонки отправляется сообщение (статус) о возможности заправки с помощью накопительной карты.

В силу сложности решения и отсутствия выборки для обучения нейронной сети, смоделируйте работу модуля следующим образом:

Данные, полученные с камер,  хранятся в файле “Camera\_load.csv”, в котором указаны даты съемки, статус распознавания - распознан/не распознан, гос. номер, имя файла с фото.

Вам необходимо предусмотреть в интерфейсе возможность проверки вашего решения:

1. Для того, чтобы окно работы с камерой стало доступным, предусмотрите в интерфейсе наличие элемента, по нажатию на который члены жюри смогут проверить смоделированную ситуацию нераспознанного гос.номера - нажать на элемент - звуковой сигнал - форма доступна для ввода гос.номера и выбора фото.
2. Для проверки правильного распознавания фото - добавьте в интерфейс элемент, по нажатию на который появится модальное окно. в этом окне жюри могут загрузить изображение, модуль выполняет распознавание госномера и выводит его на экран. Если номер распознан верно - можно подтвердить, что номер распознан верно и данные уйдут в БД. Если номер распознан неверно - фото удаляется.

*При сохранении данных вам необходимо руководствоваться правилами обмена данными, предусмотренными схемой архитектуры, представленной выше.*

**Модуль “Заправочная колонка”**

Разработайте оконное приложение для работы на АЗС с автоматической заправкой  через терминал.

Особенности - связь с HTTP сервером  
Требования к заправочной колонке:

* имеет постоянный ID;
* имеет адрес заправки (текст поле);
* при каждом запуске генерирует случайные ключ сессии (текст 12 символов);
* форма имеет текст поля для цен  (92,95,98,ДТ) и кнопку установить цены.  
  (должен посылаться запрос на ЦС с новыми ценами и адресом HTTP этого АЗС в базу данных)

Если информация о приехавшем ТС содержится в базе (получена от модуля “Камера”), ТРК предлагает вид топлива и ограничение по объему покупаемого топлива, если клиент пользовался опцией «заправка до полного бака с лимитом по объему топлива». Для начала заправки клиент должен выбрать, либо подтвердить предложенные в интерфейсе раздаточной колонки вид топлива, режим заправки, способ оплаты и произвести оплату.

Виды топлива:

- АИ-92;

- АИ-95;

- АИ-98;

- Дизельное Топливо (ДТ).

 Режимы заправки:

- на определенную сумму;

- определенное количество литров;

- до полного бака с ограничением по количеству литров.

Порядок работы колонки

При запуске пользователь должен ввести ID АЗС, номер колонки и нажать кнопку “Начать работу”. При начале работы приложение связывается с указанной АЗС по протоколу webSocket. (номер порта для связи должен быть 102хх, где вместо хх двузначный ID АЗС)

Важно! Колонка может обмениваться данными с центральным сервером (и далее с базой данных) только через АЗС.

В основном окне приложения должен быть реализован следующий пользовательский интерфейс:

*Шаг 1. Пользователь выбирает один из доступных видов топлива*Доступность вида топлива определяется наличием нужного пистолета на колонке в данный момент и наличием топлива выбранного типа в топливном хранилище.

*Шаг 2. Пользователь выбирает режим заправки*Режимы заправки:  
 - фиксированный объем;  
 - фиксированная цена;

-  до полного бака с ограничением по объему.

*Шаг 3. Выбор способа оплаты*

Способы оплаты:

- накопительной картой (данный способ доступен, если этот клиент уже заправлялся ранее и на его счету остались деньги)

- кредитной картой клиента (данный способ доступен только при если у клиента оформлена кредитная карта и кредитный лимит не исчерпан)

- банковской картой (данный способ доступен при наличии связи с банком)

*Шаг 4. Ввод данных карты (если выбрана оплата банковской картой)*

Данные банковской карты должны вводиться в модальном окне, которое содержит:

* поле ввода номера карты (формат: 16 цифр, разбитых пробелами на группы по 4 цифры)
* поле ввода “Держатель карты” (формат: фамилия и имя, разделенные пробелом)
* Кнопка “Оплатить” (становится доступна для нажатия только при соответствии данных в полях ввода указанному формату)