СБЕР УНИВЕРСИТЕТ

Курсовой проект «Автомойки»

Димиденко Максим
Емельянов Антон
Левина Анна
Полада Михаил
Старостин Сергей
Тюрников Святослав

Максим Сергеевич Димиденко
SberData
Junior Data инженер
msdimidenko@sberbank.ru

+ прошлись по всем темам Java
- последние темы про токены, docker, liquibase, kafka были сильно сжаты

Левина Анна Михайловна
ДИТ Розничный бизнес
Junior Java разработчик
fedosova.a.mi@sberbank.ru

+ структурированный учебный план (полный путь от консольного приложения до клиент-серверного)
- 4 месяца это мало

Емельянов Антон Владимирович
ДИТ Розничный бизнес
Junior Frontend разработчик
emelyanov.a.vladimiro@sberbank.ru

+ видео-лекции довольно подробно раскрывают темы

Полада Михаил Викторович
SberWorks
Junior Java разработчик
mvpolada@sberbank.ru

+ была возможность работать как самостоятельно, так и в команде
- 4 месяца это очень мало для изучения всех тем затронутых в данном курсе

Старостин Сергей Витальевич
ДИТ Корпоративно-инвестиционный бизнес
Senior инженер по сопровождению
starostin.s.vi@sberbank.ru

+ поработали со Spring и Hibernate
+ разобрали микросервисную архитектуру

Тюрников Святослав Игоревич
ДИТ Корпоративно-инвестиционный бизнес
Middle PEGA разработчик
sityurnikov@sberbank.ru

+ хороший опыт командной разработки приложения с микросервисной архитектурой
- на последние более сложные темы не хватило времени курса (хотелось бы рассмотреть более

Проект

Приложение представляет пользователям удобный функционал записи на автомобильные станции.

В нем можно просмотреть список всех станций, найти ближайшую свободную и записаться в один клик.

Основной функционал доступен как через web-приложение, так и через Telegram-бота.

Пользовательские сценарии

Я как пользователь хочу

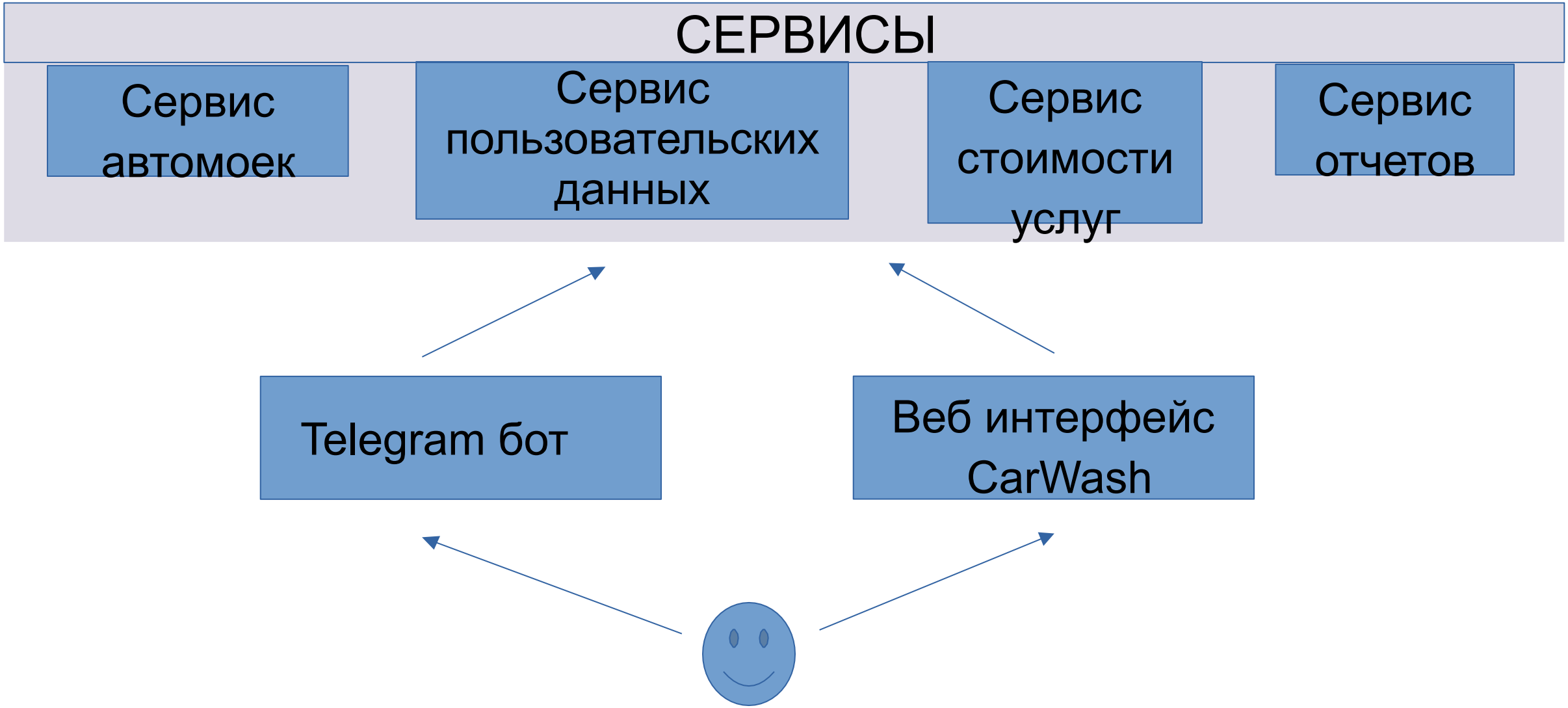
- иметь возможность записаться на ближайшую автомойку через телеграм бота
- иметь возможность отменить запись, узнать об изменении времени работы комплекса
- получить информацию о текущем статусе заказа
- иметь информацию о доступных в городе станциях обслуживания

Пользовательские сценарии

Я как администратор хочу

- разместить информацию в приложении, чтобы повысить клиентопоток
- хочу управлять жизненным циклом станции используя приложение CarWash. (начать обслуживание, завершить мойку). При этом пользователи получат уведомление о смене статуса
- оперативно узнавать статистику посещения станции
- хочу оперативно узнавать статистику по всем работающим через CarWash станциям

Концепт


СЕРВИСЫ
Сервис автомоек
Сервис пользовательских данных
Сервис стоимости услуг
Сервис отчетов
Telegram бот
Веб интерфейс CarWash

Стек технологий

- Микросервисная архитектура
- Сервисы написаны на Spring Boot
- Сервисы взаимодействуют между собой синхронно через HTTP запросы
- Модули взаимодействуют с БД (JDBC, Hibernate, Spring Data)
- Инструменты тестирования JUnit & Mockito
- Контейнеризация на базе Docker

CarWash-Service

- Сервис отвечает за хранение данных о мойках и заказах, и предоставление этой информации потребителям
- Spring Boot приложение с подключением к базе данных PostgreSQL с использованием библиотеки Hibernate
- carWash обращается за ценой в priceService
- Предоставляет информацию для reportService по запросу
- Предоставляет telegramBot список ближайших свободных моек

Ссылка на репозиторий:

API CarWash-Service (автомойки и локация)

Получить список всех автомоек (GET)
http://localhost:8080/car-wash/all
Создать автомойку (POST)
http://localhost:8080/car-wash/create
{ "title":"Мойка Красноярск ", "address":"Красноярск адрес", "latitude":"56.02", "longitude":"92.87", "ownerId":"777", "cityId":"1" }
Получить список ближайших свободных автомоек с ценами на дату (GET)
http://localhost:8080/car-wash/nearest-free-car-washes-by-date?latitude=55.6&longitude=37.7&date=2022-09-09
Удалить автомойку по id (DELETE)
http://localhost:8080/car-wash/delete-by-id?id=2
Получить список городов (GET)
http://localhost:8080/city/all
Узнать город по id мойки (GET)
http://localhost:8080/city/get-by-id?id=1
Добавить информацию о городе (POST)
http://localhost:8080/city/create
{"title":"City669"}
Удалить город (DELETE)
http://localhost:8080/city/delete-by-id?id=2 (DELETE)

API CarWash-Service (заказы)

Получить все заказы (GET)
http://localhost:8080/order/all
Создать заказ (POST)
http://localhost:8080/order/create
{ "userId":"444", "status":"В работе", "carWashId":"2", "date":"2022-09-08" }
Удалить заказ по id (DELETE)
http://localhost:8080/order/delete-by-id?id=2
Создать заказ (POST без тела)
http://localhost:8080/order/update-status?id=1&status=123
Получить заказы пользователя (GET)
http://localhost:8080/order/by-userId?id=1234
Получить все заказы с ценами за выбранный период (GET)
http://localhost:8080/order/by-dates?startDate=2022-09-10&endDate=2022-09-11
Получить все заказы с ценами за выбранный период по выбранной автомойке (GET)
http://localhost:8080/order/by-dates-by-car-wash?startDate=2022-09-09&endDate=2022-09-10&id=2

User-Service

- Сервис отвечающий за хранение информации о пользователях
- Spring Boot приложение с подключением к базе данных PostgreSQL с использованием библиотеки Hibernate
- Получает информацию от telegramBot о новом пользователе, и обрабатывает её
- С фронта получает запрос на регистрацию и обрабатывает её
- Аутентификация и авторизация пользователя

getAllCarWashUsersNames

API User-Service

Аутентификация - доступно всем без аутентификации
http://localhost:8083/user-service/api/login
Создать пользователя (POST) - доступно всем без аутентификации
http://localhost:8083/user-service/api/createCarWashUser
Изменение пользователя (PUT) - доступно всем с токеном
http://localhost:8083/user-service/api/user/updateCarWashUser
Удаление пользователя (DELETE) - доступно для роли ADMIN
http://localhost:8083/user-service/api/admin/deleteCarWashUserById/{ID пользователя}
Получить всех пользователей (GET) - доступно для роли ADMIN
http://localhost:8083/user-service/api/admin/getAllCarWashUsers
Получить данные конкретного пользователя (GET) – доступно всем с токеном
http://localhost:8083/user-service/api/user/getCarWashUserById/{userId}
Получить список имен пользователей для подготовки отчетов (GET) – доступно всем

Price-Service

- Сервис отвечающий за ценообразование товаров.
- Spring Boot приложение с подключением к базе данных PostgreSQL с использованием библиотеки Hibernate.

Ссылка на репозиторий:
https://bitbucket.org/dimmaxim21/javaschoolfinalproject/src/master/

API Price-Service (admin / user)

Вывод определенной записи
GET http://localhost:8081/user/info/1
Вывод всех записей
GET http://localhost:8081/user/getAll
Вывод всех услуг по конкретной автомойке
GET http://localhost:8081/user/getGroup/2

API Price-Service (admin)

Добавление записи

POST http://localhost:8081/admin/save Content-Type: application/json

{"idCarwash":"1","service":"Техническая мойка-экспресс","price":350}

Удаление конкретной записи

DELETE http://localhost:8081/admin/delete/1

Удаление всех записей

DELETE http://localhost:8081/admin/deleteAll

Изменение существующей записи

POST http://localhost:8081/admin/update Content-Type: application/json

{"id":"1","idCarwash":"2","service":"Техническая мойка-экспресс (updated)","price":400}

Manage prices

Add Price | Start page

Id	IdCarwash	Service	Price	Edit	Delete
1	1	Техническая мойка-экспресс	350		
2	1	Бесконтактная мойка кузова автомобиля, коврики пороги	800		
3	1	Нано (2-х фазная) мойка	1100		
4	1	Нано (3-х фазная) мойка	1780		
5	1	Чистка салона пылесосом и влажная уборка пластмассовых деталей	350		
6	1	Чистка стекол изнутри химическими средствами	350		
7	1	Полировка пластмассовых деталей салона химическими средствами	300		
17	2	Комплексная химчистка салона	15000		
18	2	Химчистка велюровых ковриков (1 шт)	300		
19	2	Химчистка потолка	5500		
20	2	Химчистка пола	4500		
21	2	Химчистка сидений	5500		
22	2	Химчистка обшивки дверей	5500		
23	2	Химчистка багажника	2100		

Report-Service

- Запрашивает данные из CarWash-Service
- Формирует отчёт и отдаёт его на фронт (за выбранные даты по конкретной мойке или всем)

Ссылка не репозиторий: https://bitbucket.org/anton_emelyanov/report_service/src/master/

API

Получить все заказы с ценами за выбранный период (GET)
http://localhost:8089/report/by-dates?startDate=2022-09-08&endDate=2022-09-09
Получить все заказы с ценами за выбранный период по конкретной автомойке
http://localhost:8089/report/by-dates-by-car-wash?startDate=2022-09-08&endDate=2022-09-09&id=1

Telegram-bot

(https://t.me/CarWashTbot)

- Отображает пользователю список свободных моек, за информацией обращается в CarWash-Service
- Предоставляет возможность пользователю зарегистрироваться на сайте, обращается в userService
- Позволяет оформить новый заказ на выбранную мойку, обращается в CarWash-Service
- Позволяет отменить заказ, обращается в CarWash-Service

Ссылка на репозиторий:

https://bitbucket.org/m_blaz/javaschoolfinalproject/src/master/

Front-end

- Обращается в CarWash-Service и получает\отправляет информацию о мойках и заказах
- Отправляет запрос на регистрацию в userService
- При построении отчётов получает\отправляет информацию в reportService
- Получает список свободных моек из CarWash-Service
- Отправляет информацию о заказе в CarWash-Service

Ссылка на репозиторий:
https://bitbucket.org/sityurnikov/carwashproject/src/master/

Резюме

1. Проект реализован согласно бизнес требованиям. Пользователи могут познакомиться с функционалом через web-интерфейс
2. Проект реализован в соответствии с архитектурой
3. Проект представляет собой микросервисную архитектуру, в которой используется REST-взаимодействия.
4. Модули покрыты unit тестами

Спасибо за внимание !