Technical Specification

* Задание: разработать Telegram бота для записи в барбершоп, реализовав основное API на NodeJS, используя различные технологии для хранения данных, а также добавив вспомогательные функции, такие как рассылка почты, резерв и восстановление данных
* Технологии: Telegram API, NodeJS, NestJS, Firebase, PostgreSQL, MongoDB, RabbitMQ, Docker (docker file, docker compose) , Git
* Возможности:

1. Записаться на услугу
   1. Ввести адрес электронной почты - для рассылки почты
   2. Выбрать услугу - барбер-шоп открыт каждый день и работает с 10 до 22, одна услуга рассчитана на 1 час, что следует учитывать при реализации API
      1. Услуга - Стрижка 1 час
      2. Услуга - Бритье 1 час
   3. Выбрать дату и время
   4. \*Выбрать барбера - дополнительное задание, в котором необходимо расширить БД и API для поддержки данной функции
2. Просмотр запланированных записей - реализовать API и команду бота
3. Просмотр истории записей - реализовать API и команду бота
4. Перенести запись на другое время, дату - реализовать API и команду бота. Также добавить рассылку письма о переносе записи
5. Сменить услугу - реализовать API, команду бота и также добавить рассылке письма о переносе записи
6. Отменить запись - реализовать API, команду бота, которая бы позволяла для начала показать список текущих записей, а потом выбрать одну из них - для отмены. В случае успешной отмены - отослать письмо
7. \* Сменить барбера - дополнительное задание, в котором необходимо расширить БД и API для поддержки данной функции

* Шаги реализации:

1. Создать необходимые для реализации проекта репозитории на базе GitHub
2. Изучить документацию по созданию telegram бота и создать собственного для дальнейшей разработки проекта <https://core.telegram.org/bots>
3. Изучить документацию по фреймворку NestJS, который будет использован для реализации проекта
4. Создать проект на NestJS и предусмотреть возможность n-layer архитектуры
5. API, база данных в PostgreSQL и база данных в MongoDB должны располагаться в разных докер контейнерах и собираться через docker compose - по ходу написания проекта
6. Спроектировать архитектуру базы данных для хранения записей в барбершоп, а затем реализовать ее.
7. Предусмотреть возможность миграций для БД
8. Написать API, заложенное описанием возможностей бота, используя NestJS, на основе реализованной ранее n-layer архитектуры и PostgreSQL, как диалект для работы с БД, изучив и применив пакет для работы с PostgreSQL
9. При работе с БД предусмотреть транзакции
10. Реализовать агрегационное представление (View) в виде - записи на текущий день - по клиентам и услугам
11. Реализовать хранимые процедуры (Stored Procedure) по сбору статистики по услугам за последний месяц
12. Изучить предоставляемые возможности Firebase
13. Изучить пакет для работы с Firebase
14. Создать проект в Firebase
15. Реализовать механизм аутентификации, используя при этом Firebase, для хранения пользователей и их токенов и основную базу данных для хранения пользователей, поддерживать их целостность и консистентность - реализовать API для работы с пользователями как для основной базы, так и базы в Firebase
16. Сохранять историю переписки с ботом в Firebase в виде логов
17. Добавить, соответствующие описанию возможностей, команды для Telegram бота - приветствие, ввод адреса электронной почты, выбор услуги, перенос записи, смена услуги, отмена записи, просмотр текущих записей, просмотр истории записей
18. Реализовать возможность создания back up данных, хранящихся в SQL БД, копируя все данные в MongoDB, используя планировщик задач, такой как cron - время и частота данной операции на свое усмотрение
19. Разобрать документацию по RabbitMQ и реализовать очередь, для реализации backup и restore данных, чтобы при случае операции завершившейся неудачно - запустить ее заново из RabbitMQ
20. Реализовать рассылки писем, предусмотренные возможностями описанными выше
21. Опубликовать API с помощью Firebase Hosting

* Список источников

1. Firebase
   1. <https://firebase.google.com/docs?authuser=0>
   2. <https://coursehunter.net/course/firebase-master-klass> (english)
2. SQL, PostgreSQL - Head FirstSQL
   1. <https://coursehunter.net/course/sql-polnoe-rukovodstvo> (english)
3. NoSQL, MongoDB - Introduction to NoSQL
   1. <http://www.ccs.neu.edu/home/kathleen/classes/cs3200/20-NoSQLMongoDB.pdf>
4. RabbitMQ
   1. <https://habr.com/ru/post/447074/>
   2. <https://habr.com/ru/post/447250/>
   3. <https://www.rabbitmq.com/tutorials/tutorial-one-javascript.html>
5. Docker Tutorial
   1. <https://docker-curriculum.com/>
6. REST API Tutorial
   1. <https://www.restapitutorial.com/>
7. Telegram API
   1. <https://core.telegram.org/bots>
   2. <https://github.com/yagop/node-telegram-bot-api>