Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и кибербезопасности
Высшая школа программной инженерии

КУРСОВОЙ ПРОЕКТ

Сервис для подборки аниме-картинок

по дисциплине «Конструирование программного обеспечения»

Выполнили:

Студенты группы 5130904/30104 Демиденко Н. Д.

Кравченко Н. В.

Лебедев А. И.

Пиявкин А. А.

Руководитель:

Ст. преподаватель Иванов А. С.

Санкт-Петербург
2025

СОДЕРЖАНИЕ

Определение проблемы

Ценителям аниме и просто любителям красивых картинок порой бывает проблематично найти источник, в котором будут собраны изображения с аниме почти на любой вкус, а также найти картинку, соответствующую какой-либо категории. Существующие решения на основе Discord ботов могут быть недоступны ряду пользователей по независящим от них причинам, а искать и открывать специализированные web-сайты не очень удобно.

Выработка требований

Пользовательские сценарии:

1) Пользователь не знает о своих желаниях и хочет получить случайную аниме-картинку.

2) Пользователь знает, чего хочет, и запрашивает картинку из определенной категории (на выбор).

3) Пользователю понравилась определенная картинка и он хочет сохранить ее в свою коллекцию.

Разработка архитектуры и детальное проектирование

Репозиторий: https://github.com/Glaz0k/kawaii-keeper

Технологический стек:

- Язык для написания серверной части: Java (Spring)
- База данных: PostgreSQL
- UI: Telegram (java-telegram-bot-api https://github.com/pengrad/java-telegram-bot-api)

Внешняя зависимость: https://nekosia.cat/

Соотношение R/W нагрузки: 90% - чтение (просмотр новых картинок), 10% - запись (сохранение понравившихся картинок в коллекцию).

Объемы трафика: 30 ГБ/месяц

Объемы дисковой системы: 100 ГБ

В связи с тем, что данный сервис подпадает под «Пакет Яровой», нам необходимо хранить переписки пользователей 1 год (3 года с 2026г). Однако поскольку на данном этапе наш сервис небольшой по охвату, о несоответствии законодательству можно не беспокоиться. Поэтому было принято решение не хранить переписку.

Конфигурация сервиса:

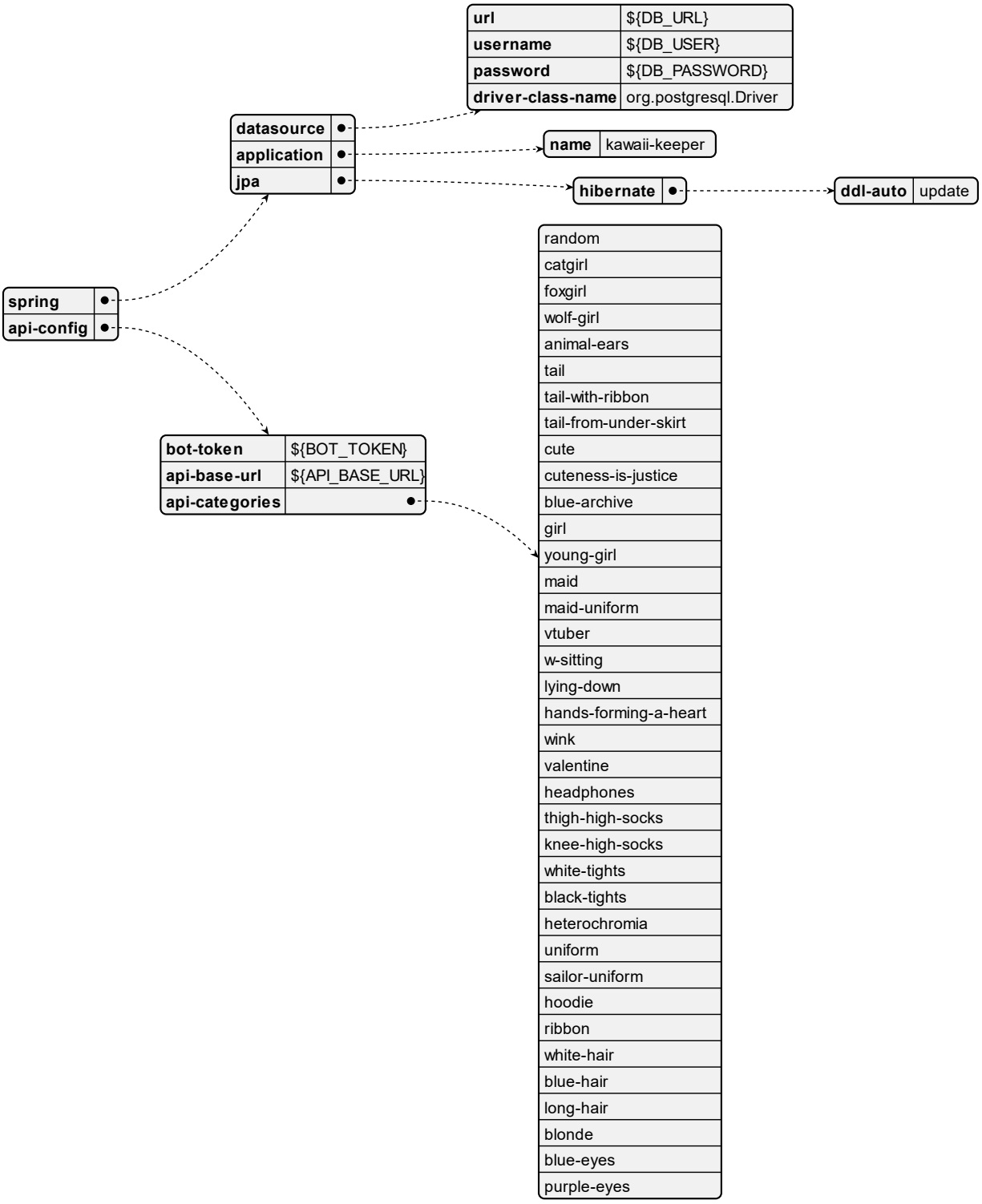


Диаграмма Level 1:

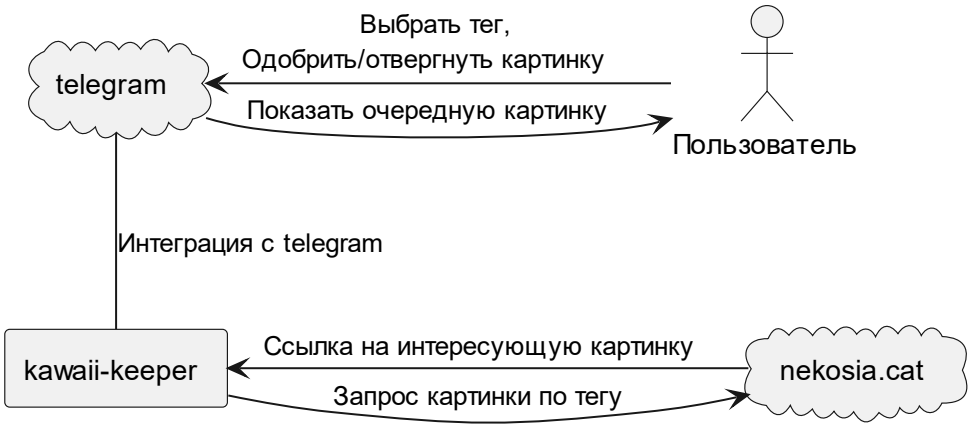


Диаграмма Level 2:

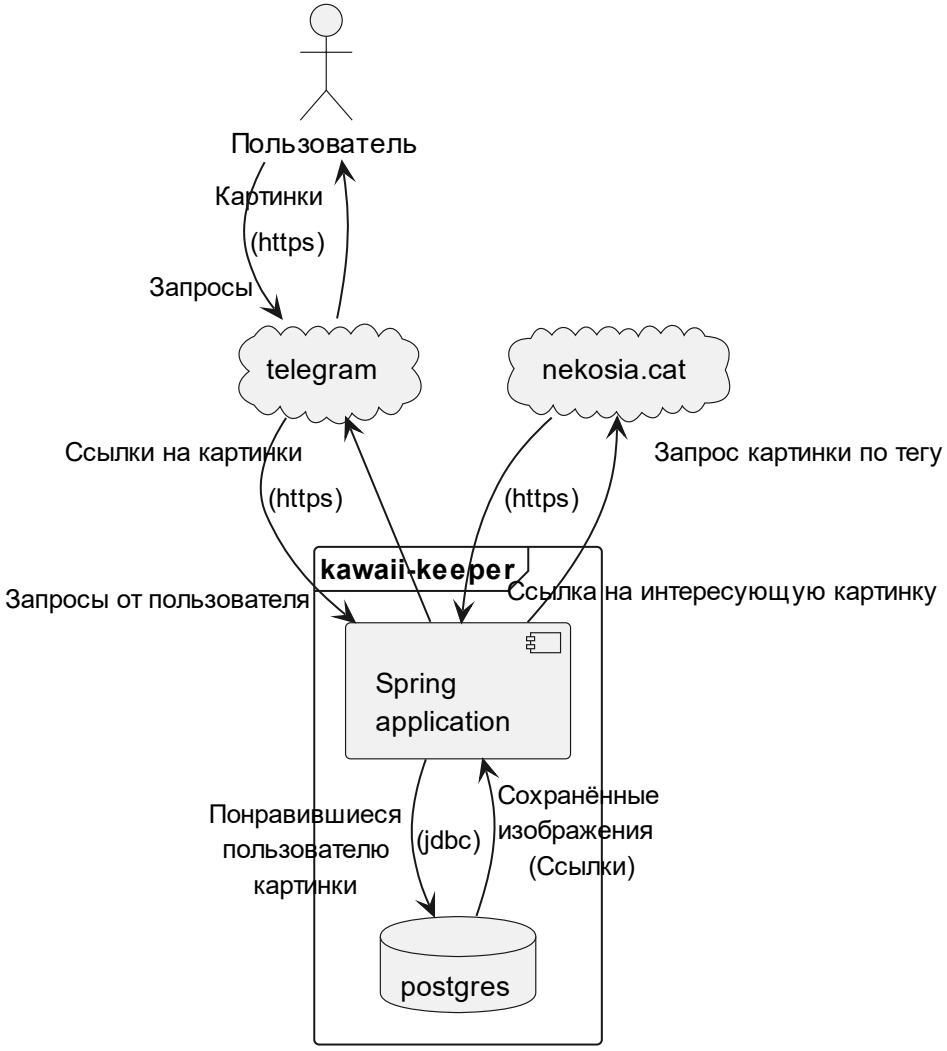
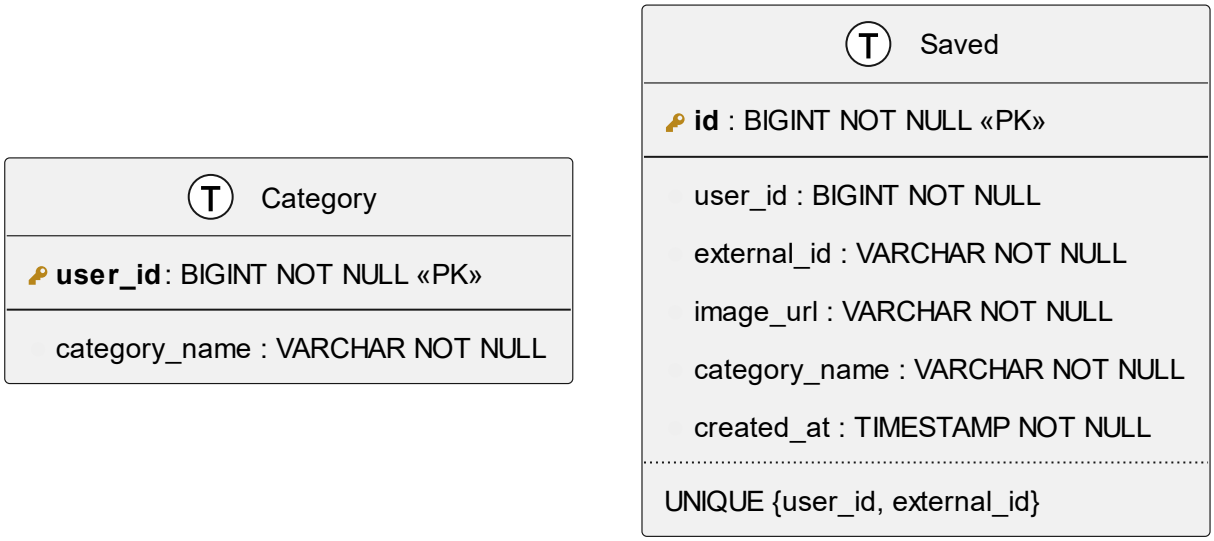


Схема базы данных:

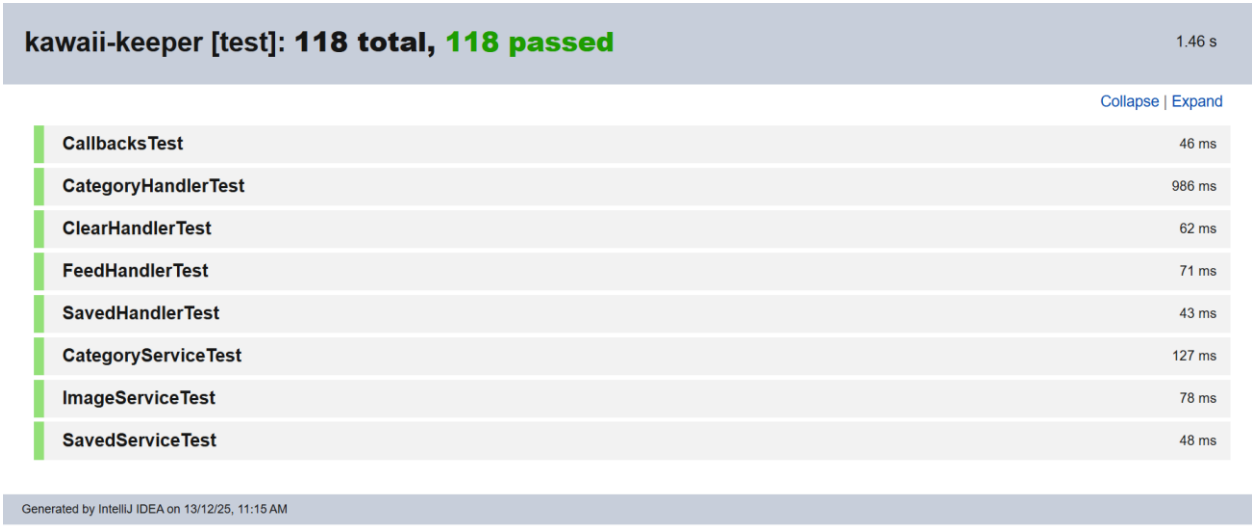


SLA (Service Level Agreement):

Простой возможности отправки запроса картинки после аварии должен быть устранен за 1 неделю. В противном случае, поскольку сервис бесплатный, принесем публичные извинения и обязуемся отправить подходящую картинку как можно скорее.

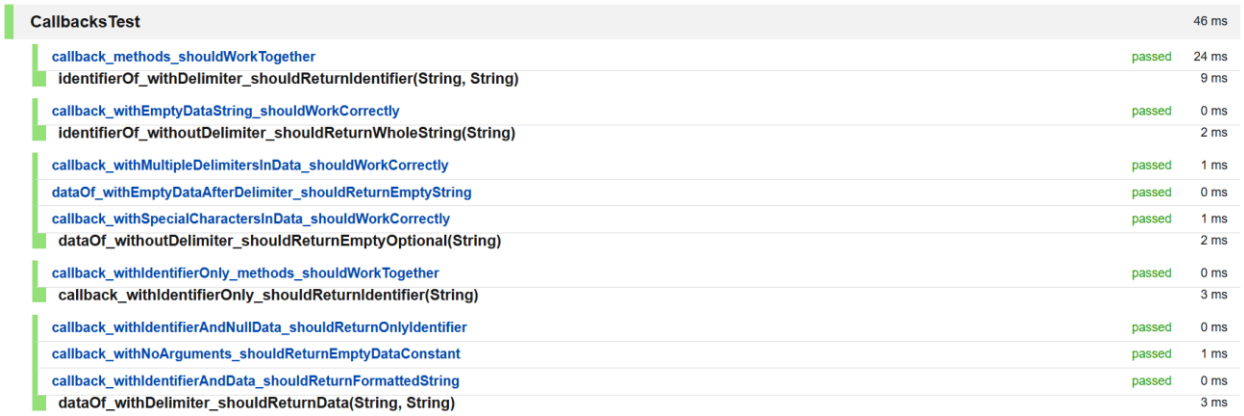
Тестирование

Общий отчет тестирования системы:



1) CallbacksTest

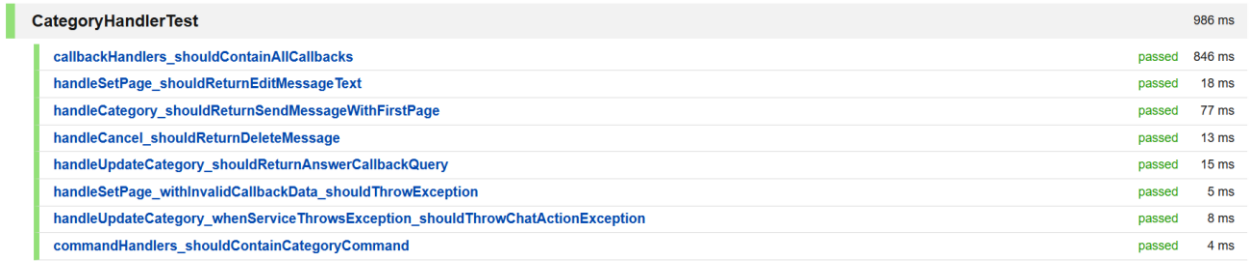
Тестирует утилитный класс для работы с callback-данными Telegram inline-кнопок.



CallbacksTest		46 ms
callback_methods_shouldWorkTogether	passed	24 ms
identifierOf_withDelimiter_shouldReturnIdentifier(String, String)		9 ms
callback_withEmptyDataString_shouldWorkCorrectly	passed	0 ms
identifierOf_withoutDelimiter_shouldReturnWholeString(String)		2 ms
callback_withMultipleDelimitersInData_shouldWorkCorrectly	passed	1 ms
dataOf_withEmptyDataAfterDelimiter_shouldReturnEmptyString	passed	0 ms
callback_withSpecialCharactersInData_shouldWorkCorrectly	passed	1 ms
dataOf_withoutDelimiter_shouldReturnEmptyOptional(String)		2 ms
callback_withIdentifierOnly_methods_shouldWorkTogether	passed	0 ms
callback_withIdentifierOnly_shouldReturnIdentifier(String)		3 ms
callback_withIdentifierAndNullData_shouldReturnOnlyIdentifier	passed	0 ms
callback_withNoArguments_shouldReturnEmptyDataConstant	passed	1 ms
callback_withIdentifierAndData_shouldReturnFormattedString	passed	0 ms
dataOf_withDelimiter_shouldReturnData(String, String)		3 ms

2) CategoryHandlerTest

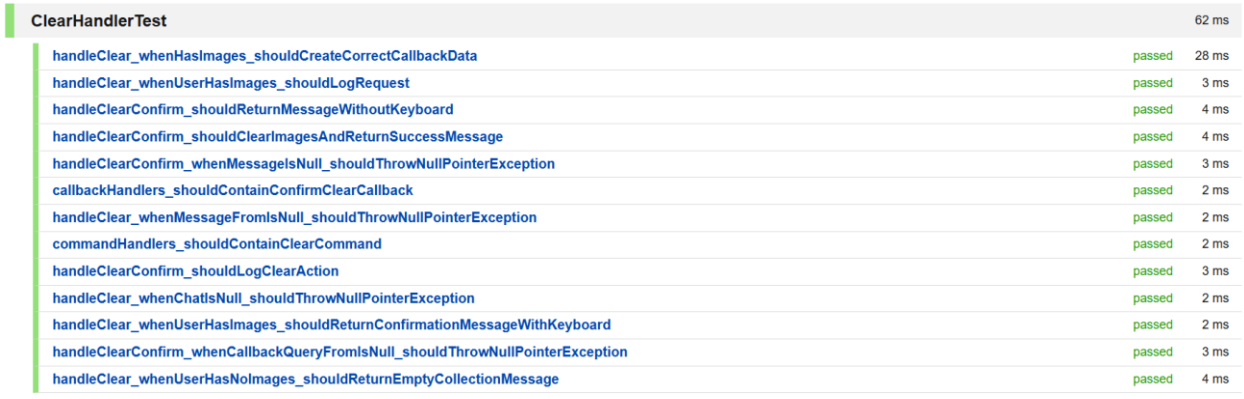
Тестирует обработчик выбора пользователем категорий изображений.



CategoryHandlerTest		986 ms
callbackHandlers_shouldContainAllCallbacks	passed	846 ms
handleSetPage_shouldReturnEditMessageText	passed	18 ms
handleCategory_shouldReturnSendMessageWithFirstPage	passed	77 ms
handleCancel_shouldReturnDeleteMessage	passed	13 ms
handleUpdateCategory_shouldReturnAnswerCallbackQuery	passed	15 ms
handleSetPage_withInvalidCallbackData_shouldThrowException	passed	5 ms
handleUpdateCategory_whenServiceThrowsException_shouldThrowChatActionException	passed	8 ms
commandHandlers_shouldContainCategoryCommand	passed	4 ms

3) ClearHandlerTest

Тестирует очистку коллекции сохранённых изображений.



ClearHandlerTest		62 ms
handleClear_whenHasImages_shouldCreateCorrectCallbackData	passed	28 ms
handleClear_whenUserHasImages_shouldLogRequest	passed	3 ms
handleClearConfirm_shouldReturnMessageWithoutKeyboard	passed	4 ms
handleClearConfirm_shouldClearImagesAndReturnSuccessMessage	passed	4 ms
handleClearConfirm_whenMessageIsNull_shouldThrowNullPointerException	passed	3 ms
callbackHandlers_shouldContainConfirmClearCallback	passed	2 ms
handleClear_whenMessageFromIsNull_shouldThrowNullPointerException	passed	2 ms
commandHandlers_shouldContainClearCommand	passed	2 ms
handleClearConfirm_shouldLogClearAction	passed	3 ms
handleClear_whenChatIsNull_shouldThrowNullPointerException	passed	2 ms
handleClear_whenUserHasImages_shouldReturnConfirmationMessageWithKeyboard	passed	2 ms
handleClearConfirm_whenCallbackQueryFromIsNull_shouldThrowNullPointerException	passed	3 ms
handleClear_whenUserHasNoImages_shouldReturnEmptyCollectionMessage	passed	4 ms

4) FeedHandlerTest

Тестирует ленту изображений

FeedHandlerTest		71 ms
handleStart_whenNewUser_shouldReturnGreetingsAndFeed	passed	32 ms
handleNext_whenUserHasNoCategory_shouldThrowChatActionException	passed	6 ms
handleStart_whenExistingUser_shouldReturnFeedOnly	passed	6 ms
handleSave_shouldSaveImageAndReturnAnswerAndEditKeyboard	passed	5 ms
callbackHandlers_shouldContainNextAndSaveCallbacks	passed	3 ms
handleNext_shouldReturnEditMediaAndCaption	passed	6 ms
handleSave_whenServiceThrowsException_shouldThrowChatActionException	passed	4 ms
commandHandlers_shouldContainStartCommand	passed	2 ms
handleSave_withInvalidCallbackData_shouldThrowException	passed	3 ms
handleStart_whenExistingUserButNoCategory_shouldThrowException	passed	4 ms

5) SavedHandlerTest

Тестирует просмотр пользователем сохранённой коллекции изображений.

SavedHandlerTest		43 ms
handleRemove_whenRemovingFirstImageOfTwo_shouldShowSecondImage	passed	5 ms
handleSetPage_shouldReturnEditMessageMediaWithKeyboard	passed	3 ms
handleRemove_whenRemovingLastImage_shouldDeleteMessageAndSendNotFound	passed	4 ms
handleRemove_whenRemovingNotLastImage_shouldEditMessageWithRemaining	passed	3 ms
handleRemove_whenServiceThrowsException_shouldThrowChatActionException	passed	4 ms
handleSaved_whenUserHasImages_shouldReturnSendPhotoWithKeyboard	passed	4 ms
commandHandlers_shouldContainSavedCommand	passed	2 ms
handleSaved_whenUserHasNoImages_shouldReturnNotFoundMessage	passed	3 ms
callbackHandlers_shouldContainSetPageAndRemoveCallbacks	passed	3 ms
handleRemove_whenInvalidCallbackData_shouldThrowChatActionException	passed	4 ms
handleSetPage_whenInvalidCallbackData_shouldThrowChatActionException	passed	4 ms
handleSetPage_whenServiceThrowsException_shouldThrowChatActionException	passed	4 ms

6) CategoryServiceTest

Тестирует бизнес-логику работы с категориями.

CategoryServiceTest		127 ms
updateCategory_shouldOverwriteExistingCategory	passed	97 ms
hasCategory_whenCategoryDoesNotExist_shouldReturnFalse	passed	2 ms
updateCategory_whenConfigHasEmptyList_shouldAlwaysThrowException	passed	3 ms
setDefaultCategory_withZeroUserId_shouldWork	passed	2 ms
updateCategory_withCaseSensitiveCategoryName_shouldRespectCase	passed	2 ms
findCategory_whenCategoryExists_shouldReturnCategory	passed	1 ms
updateCategory_shouldSaveCategoryWithNewName	passed	2 ms
setDefaultCategory_shouldSaveNewCategoryWithDefaultName	passed	2 ms
hasCategory_whenCategoryExists_shouldReturnTrue	passed	2 ms
getDefaultCategoryName_whenConfigHasSingleCategory_shouldReturnIt	passed	2 ms
getDefaultCategoryName_shouldReturnFirstCategoryFromConfig	passed	2 ms
updateCategory_whenCategoryNameIsNull_shouldThrowNullPointerException	passed	3 ms
getDefaultCategoryName_whenConfigIsEmpty_shouldThrowIndexOutOfBounds	passed	2 ms
updateCategory_whenCategoryNameNotInConfig_shouldThrowRuntimeException	passed	2 ms
updateCategory_whenCategoryNameIsEmptyString_shouldCheckConfig	passed	1 ms
hasCategory_withZeroUserId_shouldWork	passed	1 ms
findCategory_whenCategoryDoesNotExist_shouldReturnEmptyOptional	passed	1 ms

7) ImageServiceTest

Тестирует интеграцию с внешним API изображений.

ImageServiceTest		78 ms
getByExternalId_withEmptyCategoryName_shouldWork	passed	49 ms
pollNext_whenJsonMissingFields_shouldThrowRuntimeException	passed	3 ms
getByExternalId_shouldMakeCorrectRequestAndParseResponse	passed	2 ms
pollNext_whenApiReturnsNotFound_shouldThrowRuntimeException	passed	4 ms
getByExternalId_whenJsonMissingRequiredFields_shouldThrowRuntimeException	passed	3 ms
pollNext_whenJsonResponseIsInvalid_shouldThrowRuntimeException	passed	2 ms
pollNext_whenApiReturnsError_shouldThrowRuntimeException	passed	2 ms
pollNext_shouldAcceptJsonWithExtraFields	passed	4 ms
getByExternalId_whenApiReturnsError_shouldThrowRuntimeException	passed	2 ms
pollNext_shouldMakeCorrectRequestAndParseResponse	passed	2 ms
pollNext_shouldParseComplexJsonStructureCorrectly	passed	3 ms
getByExternalId_withSpecialCharacters_shouldEncodeUrlCorrectly	passed	2 ms

8) SavedServiceTest

Тестирует сервис работы с сохранёнными изображениями.

SavedServiceTest		48 ms
saveImage_withExternalId_shouldHandleImageServiceResponse	passed	32 ms
clearImages_withZeroUserId_shouldWork	passed	0 ms
hasImages_withZeroUserId_shouldCheckRepository	passed	1 ms
saveImage_shouldNotSetIdBeforeSaving	passed	1 ms
saveImage_withImageDto_shouldCreateCorrectSavedEntity	passed	2 ms
integration_saveAndFind_shouldWorkTogether	passed	2 ms
saveImage_withImageDto_shouldSaveCorrectly	passed	1 ms
findOrderedImages_whenNoImages_shouldReturnEmptyList	passed	0 ms
clearImages_shouldCallRepositoryDeleteByUserId	passed	0 ms
findOrderedImages_shouldReturnOrderedList	passed	2 ms
removeImage_shouldCallRepositoryDelete	passed	1 ms
saveImage_withExternalId_shouldGetImageAndSave	passed	1 ms
hasImages_whenNoImages_shouldReturnFalse	passed	1 ms
findOrderedImages_shouldMapAllFieldsCorrectly	passed	1 ms
hasImages_whenImagesExist_shouldReturnTrue	passed	1 ms
removeImage_withZeroId_shouldWork	passed	1 ms
saveImage_withExternalId_whenImageServiceThrowsException_shouldPropagate	passed	1 ms

Сборка

Для сборки проекта используется docker-compose. При запуске сервиса поднимаются два контейнера:

1)Backend

2)PostgreSQL

Для запуска проекта необходимо прописать:

```
docker compose up -d
```