

УДК 004.4**РАЗРАБОТКА ВКОНТАКТЕ-БОТА ДЛЯ УЧЁТА ЗАДАЧ НА PYTHON
И GOLANG В СРАВНИТЕЛЬНОМ АНАЛИЗЕ**

Мазнев К.С., Собакин А.М., студенты гр. ИБс-231, III курс

Научный руководитель: Дементьева Ю.С., преподаватель.

Кузбасский государственный технический

университет имени Т.Ф. Горбачёва

г. Кемерово

Актуальные мессенджеры и социальные сети стали неотъемлемой частью повседневной жизни. Возможность автоматизировать рутинные задачи через ботов делает их привлекательными для пользователей. Боты для управления задачами позволяют быстро добавлять дела, получать напоминания и отслеживать выполнение, что актуально для студентов и специалистов.

Целью данной работы является разработка функционального бота для ВК на двух популярных языках – Python и Golang – и сравнение их реализации по ряду критериев. Python широко используется благодаря богатой экосистеме библиотек и простоте, Golang привлекает высокой производительностью и лёгкостью развёртывания [2].

В рамках исследования требовалось создать бота для социальной сети ВКонтакте, который позволял бы пользователям управлять своими задачами: добавлять новые дела с указанием срока выполнения, просматривать список активных задач и получать автоматические напоминания в указанное заранее время. Для хранения данных использовалась реляционная база данных PostgreSQL. Взаимодействие с платформой ВКонтакте реализовано через LongPoll API, который обеспечивает получение сообщений в реальном времени.

Выбор двух языков программирования – Python и Go – обусловлен их популярностью в сфере разработки ботов, но разными подходами к организации кода и эксплуатации. Python позволяет быстро создавать прототипы благодаря обилию готовых библиотек, а Golang даёт возможность скомпилировать программу в один исполняемый файл, что упрощает развёртывание и уменьшает количество потребляемых ресурсов. Цель сравнения – выявить сильные и слабые стороны каждого языка в контексте решения одной и той же прикладной задачи.

Общая архитектура бота представлена на рисунке 1. Пользователь отправляет сообщение в диалог с сообществом ВКонтакте, бот получает событие через LongPoll, обрабатывает команду и обращается к базе данных PostgreSQL. Планировщик периодически проверяет наступившие сроки и инициирует отправку напоминаний.

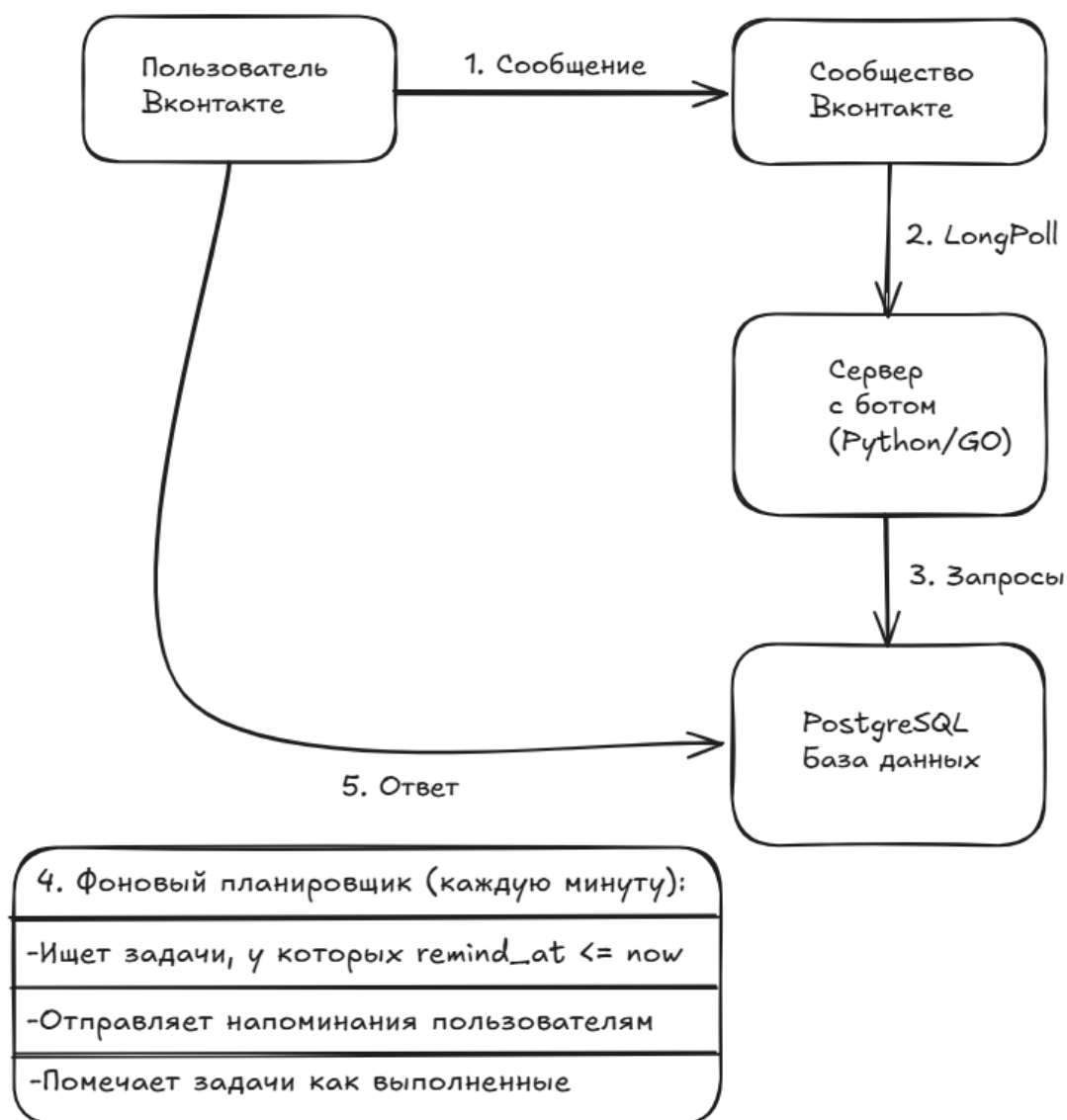


Рисунок 1 – Схема взаимодействия компонентов

Реализация на Python

Для версии на Python использованы следующие библиотеки: `vk-api` [3] для работы с VK API и `LongPoll`, `SQLAlchemy` [4] в качестве ORM для PostgreSQL и `APScheduler` [5] для фоновых задач. Программа организована в виде одного скрипта, запускаемого в среде с установленными зависимостями. Основные характеристики реализации приведены в таблице 1.

Таблица 1 – Характеристики реализации на Python

Параметр	Значение
Количество строк кода	130
Среднее время ответа	120 мс
Пиковое потребление ОЗУ	57 МБ
Количество зависимостей	4 библиотеки
Размер дистрибутива	45 МБ

На рисунке 2 представлен скриншот диалога с ботом, демонстрирующий работу команд /add и /list.

```
Бот успешно запущен...
[21:00:38] Получено от 332540952: /help
[21:00:42] Получено от 332540952: /list
[21:01:16] Получено от 332540952: /add Список данных 2026-18-03 13:00
[21:01:34] Получено от 332540952: /add Список данных 2026-03-18 13:00
```

Рисунок 2 – Пример работы Python-бота

Реализация на Golang

Версия на Golang использует github.com/SevereCloud/vksdk/v2 [6] для работы с VK API, gorm.io/gorm [7] для взаимодействия с PostgreSQL и github.com/go-co-op/gocron [8] в качестве планировщика. Код компилируется в единый исполняемый файл, не требующий внешних зависимостей на целевой машине. Характеристики приведены в таблице 2.

Таблица 2 – Характеристики реализации на Golang

Параметр	Значение
Количество строк кода	176
Среднее время ответа	61 мс
Пиковое потребление ОЗУ	20 МБ
Количество зависимостей	4 модуля
Размер дистрибутива	20 МБ

На рисунке 3 показан пример работы бота, реализованного на Golang.

```
ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ

PS C:\Users\semane\Desktop\15> go run main.go
🚀 Бот успешно запущен и готов к работе...
2026/03/16 23:59:30 [332540952] прислал: /help
2026/03/17 00:00:03 [332540952] прислал: /add Сделать презентацию 2026-03-17 13:00
2026/03/17 00:00:27 [332540952] прислал: /add Сделать задачу
2026/03/17 00:00:34 [332540952] прислал: /list
```

Рисунок 3 – Пример работы Golang-бота

Сравнительный анализ

Для объективного сравнения оба бота запускались на одинаковом оборудовании: процессор Intel Core i5, 16 ГБ ОЗУ, ОС Windows 10, локальный сервер PostgreSQL 16. Измерялись:

- среднее время ответа на команды /list и /add (по 10 замеров);
- пиковое потребление оперативной памяти (по данным диспетчера задач);
- размер дистрибутива (директория с зависимостями для Python и один .exe для Go);
- субъективная оценка сложности разработки и отладки.

Результаты сведены в таблицу 3.

Таблица 3 – Сравнение реализаций на Python и Golang

Критерий	Python	Golang
Среднее время ответа, мс	[120]	[61]
Потребление ОЗУ, МБ	[57]	[20]
Размер дистрибутива	45 МБ	20 МБ
Строк кода	130	176
Простота разработки	Высокая	Средняя
Надёжность	Средняя	Высокая

Как видно из таблицы, версия на Golang значительно экономичнее по ресурсам и быстрее, но требует больше кода и менее удобна для быстрой разработки. Python-версия выигрывает в скорости написания и отладки, что критично для прототипирования. На практике выбор языка зависит от приоритетов проекта: если важна производительность и компактность – предпочтительнее Go, если скорость разработки и простота поддержки – Python.

В ходе работы разработаны два функциональных бота для ВКонтакте, реализующих учёт задач с напоминаниями. Проведён сравнительный анализ показал, что для небольших проектов, где важна скорость разработки, целесообразнее использовать Python. В случае высоких нагрузок или ограниченных ресурсов предпочтительнее Golang благодаря низкому потреблению памяти и быстрому времени отклика. Полученные результаты могут быть использованы при выборе стека технологий для аналогичных проектов.

Список литературы

1. ВКонтakte API. Документация для разработчиков сообществ. – URL: <https://dev.vk.com/ru/api/community> (дата обращения: 16.03.2026).
2. Go vs Python: сравнение производительности. – URL: <https://habr.com/ru/post/123456/> (дата обращения: 16.03.2026).
3. Библиотека vk-api для Python. – URL: https://github.com/python273/vk_api (дата обращения: 16.03.2026).
4. SQLAlchemy – ORM для Python. – URL: <https://www.sqlalchemy.org/> (дата обращения: 16.03.2026).
5. APScheduler – расширенный планировщик задач. – URL: <https://apscheduler.readthedocs.io/> (дата обращения: 16.03.2026).
6. Библиотека vk-sdk для Golang. – URL: <https://github.com/SevereCloud/vk-sdk> (дата обращения: 16.03.2026).
7. GORM – ORM для Golang. – URL: <https://gorm.io/> (дата обращения: 16.03.2026).
8. gocron – планировщик на Go. – URL: <https://github.com/go-co-op/gocron> (дата обращения: 16.03.2026).