

УДК 622

Разработка ПО для управления бизнес-процессами в компьютерном клубе «KiberPride»

Власенков И.А., студент гр. ПИБ-221, IV курс
Научный руководитель Колокольников А.И., доцент
Кузбасский государственный технический университет
имени Т.Ф. Горбачёва, г. Кемерово

Современные предприятия сферы услуг сталкиваются с необходимостью автоматизации внутренних бизнес-процессов с целью повышения эффективности работы, снижения ошибок и улучшения качества обслуживания клиентов. Особенно это актуально для компьютерных клубов, где требуется постоянный контроль финансовых операций, учет смен сотрудников, фиксация расходов и формирование отчетности.

Компьютерный клуб «KiberPride» представляет собой организацию, в которой ежедневно выполняется большое количество операций: открытие и закрытие смен, учет наличных и безналичных средств, фиксация расходов, а также контроль работы администраторов. При отсутствии автоматизированной системы эти процессы выполняются вручную, что приводит к увеличению временных затрат и вероятности ошибок.

Целью данной работы является разработка программного обеспечения для автоматизации управления бизнес-процессами компьютерного клуба «KiberPride». В данной статье будут изложены основные функции системы, принцип взаимодействия с пользователем, преимущества использования данной системы как наиболее удобный способ хранения информации.

Особое значение в рамках автоматизации приобретает вопрос удобства взаимодействия пользователя с системой. Использование чат-ботов в качестве интерфейса является одним из наиболее перспективных направлений, поскольку такие решения не требуют дополнительного обучения персонала и могут быть интегрированы в привычные для пользователей платформы. Это позволяет значительно снизить порог входа и ускорить процесс внедрения системы. В нашем же случае, чат-бот будет реализован в российском мессенджере VK.

Разрабатываемое ПО предназначено для ведения учета денежных средств в кассе, сбора информации по всем проходящим тратам клуба. Так же бот реализует отправку информации на сервер, что позволяет хранить все отчёты и просматривать их в любой момент времени, благодаря чему старший администратор способен проверять и контролировать некоторые процессы в клубе находясь в любом месте.

Самому пользователю доступны следующие возможности:

- Открытие/закрытие смены
- Создание пересчета кассы неограниченное количество раз за смену
- Добавление трат за смену
- Просмотр отчёта на момент последнего пересчета

Данное решение позволяет избавить работников от сложности хранить информацию просто в сообщениях, теперь вся информация о смене будет попадать на сервер, где будут доступны все операции за все смены. Так же в отчете всегда будет видно суммарное количество денежных средств в кассе, что позволит нам строго контролировать заполненность кассы и в случае достижения суммы 10 тысяч рублей, касса инкассируется.

Чат-бот был создан посредством языка программирования Python в среде программирования PyCharm. При создании бота использовалась СУБД SQLite.

Для формализации процессов пересчёта были разработаны диаграммы деятельности, отражающие последовательность проведения смены в момент эксплуатации пользователем в системе. Диаграммы были созданы с помощью Progr@m4you.



Рисунок 1. Диаграмма деятельности бизнес-процесса "Пересчет кассы"

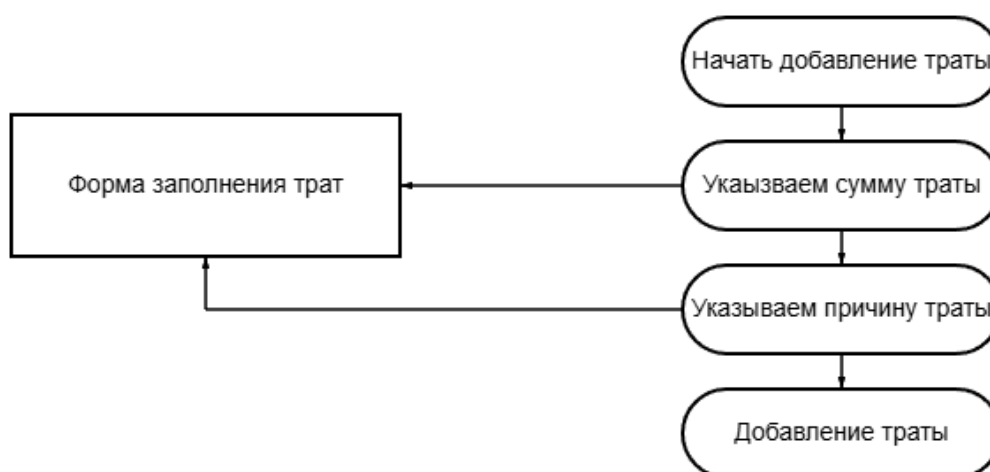


Рисунок 2. Диаграмма деятельности бизнес-процесса "Добавление трат"

Процесс создания и активации чат-бота в VK включает в себя регистрацию сообщества, настройку доступа к сообщениям, получения ключа API и подключение к серверной части, обеспечивающей обработку пользовательских запросов. После этого выполняется настройка команд, сценариев взаимодействия и механизмов обмена данными между ботом и информационной системой.

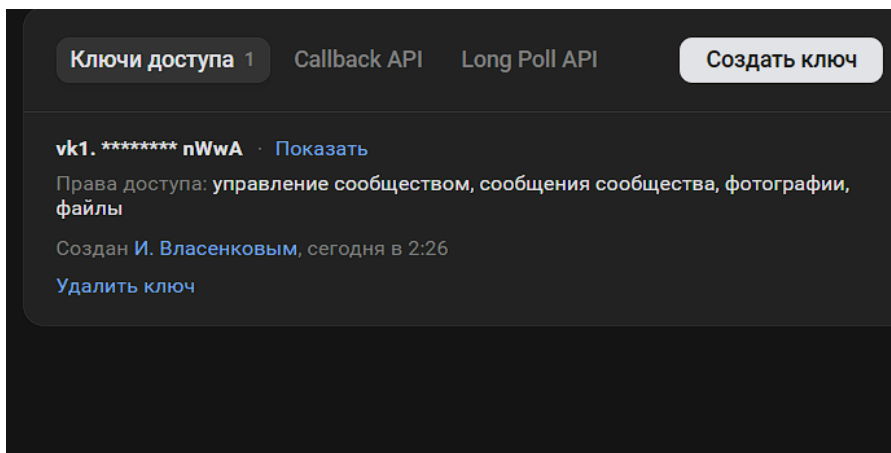


Рисунок 3. Создание ключа доступа.

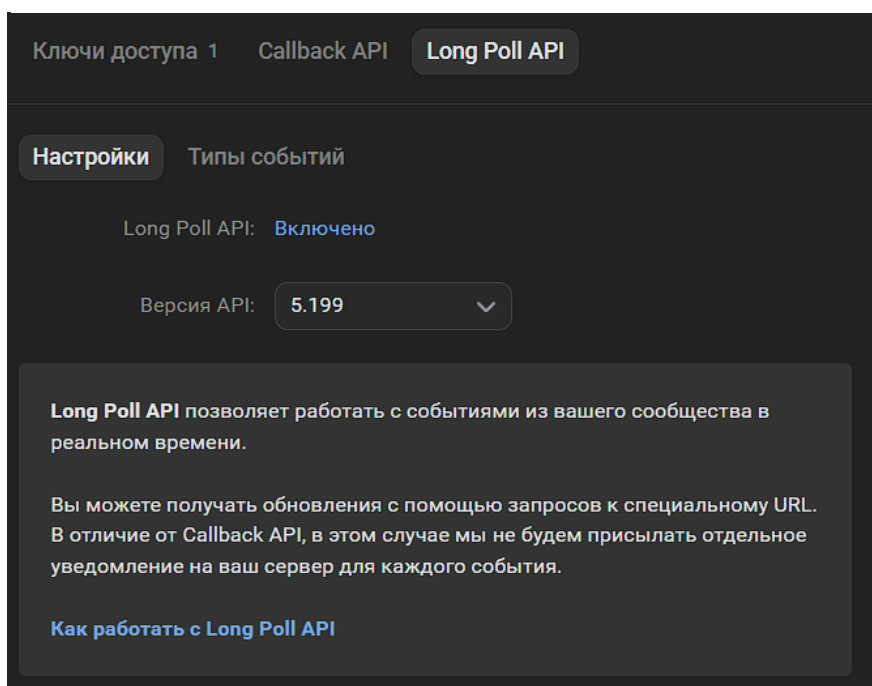


Рисунок 4. Подключение к API.

Само подключение токена происходит в одном из файлов проекта разработки. Далее, из этого файла мы уже будем импортировать информацию о токене в файл конфигурации, который и будет устанавливать связь с «ВКонтакте».

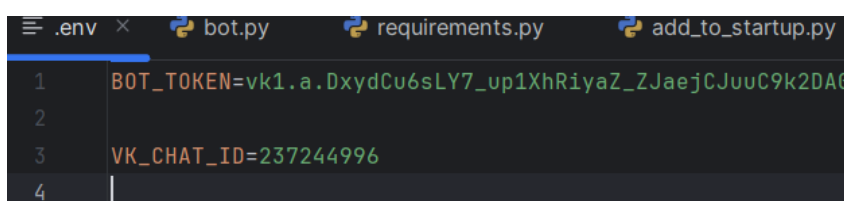
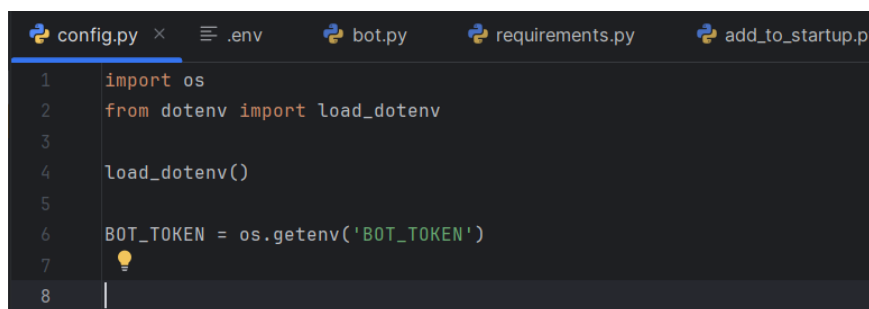


Рисунок 5. Файл с информацией о токене.



```
config.py x .env bot.py requirements.py add_to_startup.py
1 import os
2 from dotenv import load_dotenv
3
4 load_dotenv()
5
6 BOT_TOKEN = os.getenv('BOT_TOKEN')
7
8
```

Рисунок 6. Файл конфигурации

```
bot.py x run_hidden_simple.py config.py .env
9 class VKCashBot: 1 usage
211 def handle_expense_reason(self, user_id, text): 1 u
213
214     self.db.add_expense({
215         'shift_id': shift['id'],
216         'admin_name': shift['admin_name'],
217         'amount': self.user_data[user_id]['amount'],
218         'description': text,
219         'expense_date': date.today().isoformat(),
220         'expense_time': datetime.now().time().isofo
221     })
222
223     self.user_states[user_id] = "MAIN_MENU"
224     self.send(user_id, text="✅ Трата сохранена")
225     self.show_main_menu(user_id)
226
227
228 if __name__ == "__main__":
229     import os
230     from dotenv import load_dotenv
231
232     load_dotenv()
233     token = os.getenv("BOT_TOKEN")
234
235     bot = VKCashBot(token)
236     bot.run()
```

Рисунок 7. Файл запуска бота

Механизм поведения описан выше в программе. Был реализован функционал благодаря языку программирования Python. После старта бота с помощью файла запуска, его функции работают при обращении в личные сообщения сообщества, которое мы создали.

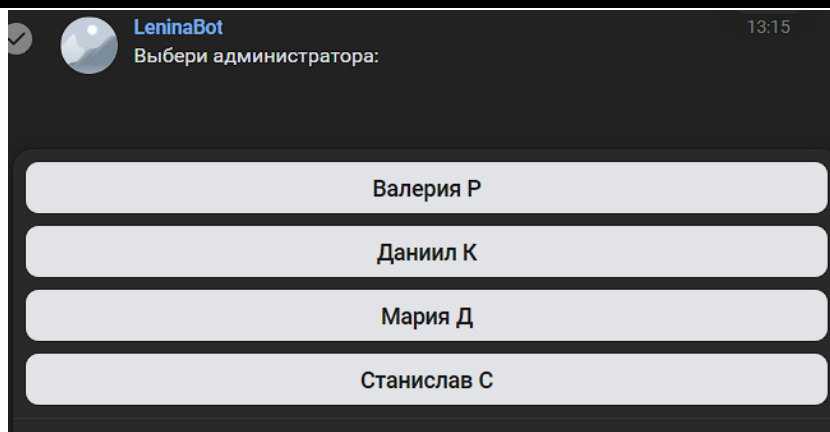


Рисунок 8. Открытие смены с ботом

После нажатия кнопки «Открыть смену», запускается сценарий открытия смены. Он достаточно короткий, чтобы описать его только в данном блоке. На деле же мы открываем смену и выбираем имя администратора на смене, больше ничего не требуется, поэтому перейдем сразу к пересчету кассы.

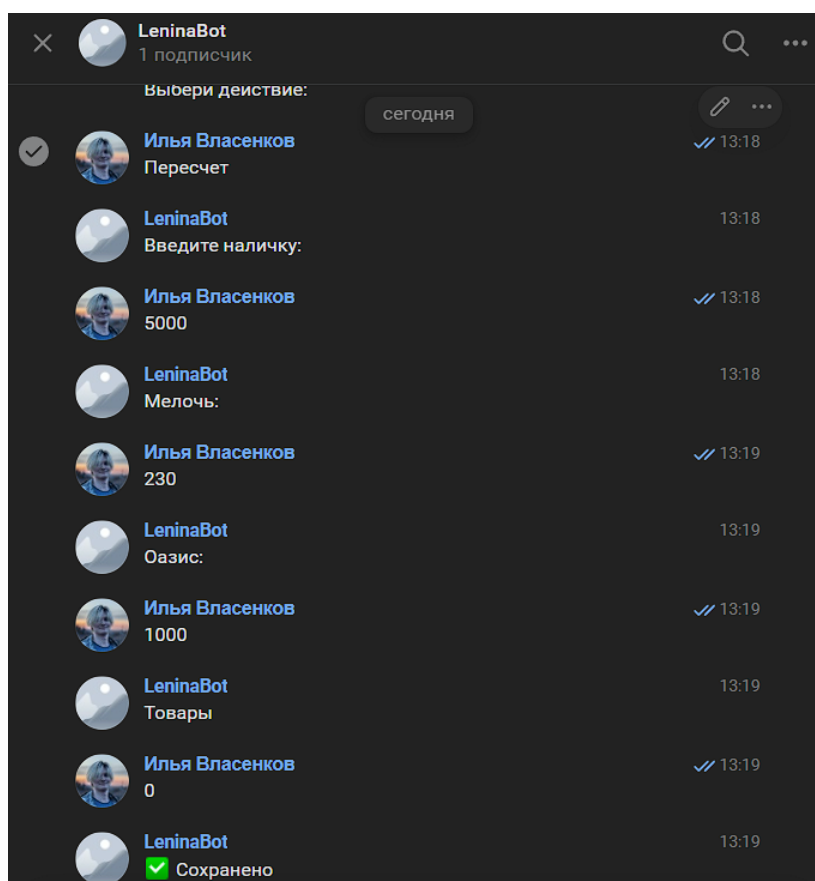


Рисунок 9. Результат работы бота в диалоге "Пересчет кассы"

На данном этапе происходит то, что было описано выше. При пересчете у нас открывается форма пересчета, в которую нам надо вписать необходимые данные.

В конце пересчета мы видим, что бот сохраняет наш пересчет, что означает, он уже находится в базе данных и старший администратор без труда сможет просмотреть его, если потребуется и сверить с пересчетом при закрытии своей смены.

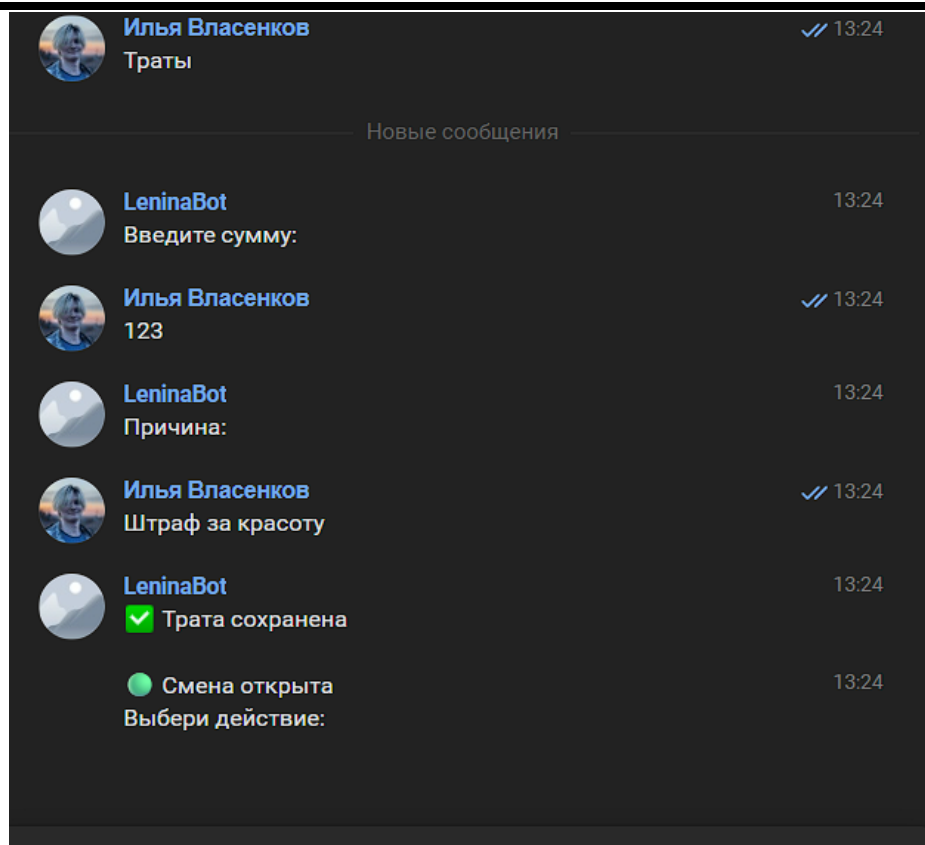


Рисунок 10. Результат работы бота в диалоге "Добавить трату"

На данном этапе происходит опять-таки то, что было описано выше. Наш чат-бот может сохранять траты, и они так же находятся в нашей базе данных.

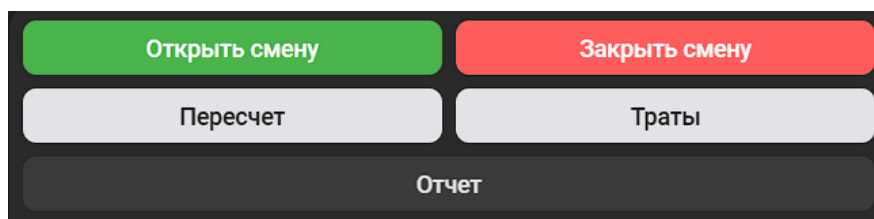


Рисунок 11. Кнопки для взаимодействия с ботом

На данной панели указаны все доступные нам кнопки и единственная, о которой я ещё не рассказал – Отчёт. Отчет у нас суммирует наличные + мелочь и вычитает траты, т.е. показывает сколько всего денег в кассе сейчас.

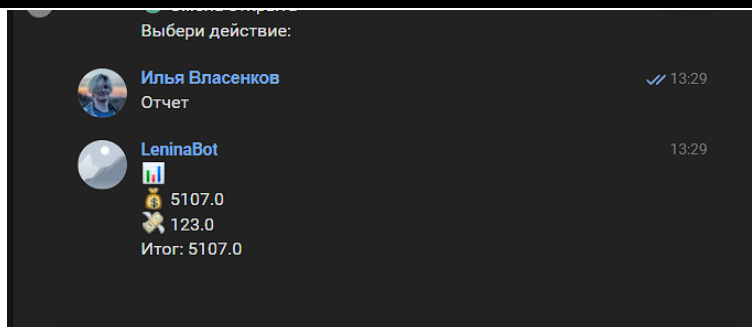


Рисунок 12. Результат работы бота в диалоге "Отчет"

В рамках данной работы был разработан чат-бот, предназначенный для управления бизнес-процессами компьютерного клуба «KiberPride». Разработанное ПО позволяет автоматизировать процесс управления финансовыми подсчётами в клубе и проверять их точность при помощи просмотров старых пересчетов.

В дальнейшем данный чат-бот будет только улучшаться, будут дорабатываться старые функции и добавляться новые. На данный момент процесс разработки направлен на создание детализации пересчетов и возможность просматривать все пересчеты и менять их, не заглядывая в базу данных, а также продолжать в чате.

Список литературы

1. Python Software Foundation. Python 3 Documentation [Электронный ресурс]. – режим доступа: <https://docs.python.org/>
2. VK. Официальная документация VK API [Электронный ресурс]. – Режим доступа: <https://dev.vk.com/>
3. Синицын С.В. Разработка чат-ботов на Python: практическое руководство. – Москва: ДМК Пресс, 2021. – 256 с.
4. SQLite3 — DB-API 2.0 interface for SQLite [Электронный ресурс]. – Режим доступа: [sqlite3 — DB-API 2.0 interface for SQLite databases — Python 3.14.3 documentation](#)
5. Datetime — Basic date and time types [Электронный ресурс]. – режим доступа: [datetime — Basic date and time types — Python 3.14.3 documentation](#)