

ОПТИМИЗАЦИЯ ОРГАНИЗАЦИОННО-АДМИНИСТРАТИВНЫХ ПРОЦЕССОВ В ШКОЛЕ СРЕДСТВАМИ ЦИФРОВЫХ ТЕХНОЛОГИЙ: ОПЫТ РАЗРАБОТКИ И ВНЕДРЕНИЯ TELEGRAM-БОТА ДЛЯ СИСТЕМЫ ШКОЛЬНОГО ПИТАНИЯ

*Кронин Лев Григорьевич,
учащийся 11 класса А*

Государственное бюджетное общеобразовательное учреждение лицей № 373

*Ляпустина Ольга Юрьевна,
учитель информатики,*

Государственное бюджетное общеобразовательное учреждение лицей № 373

Аннотация

В данной статье рассматривается опыт создания и внедрения программного решения для автоматизации процесса подачи заявок на питание в школьной столовой. Описывается полный цикл работы над проектом: от выявления проблематики и анализа существующих дефицитов в организации школьного быта до технической реализации на языке программирования Python и внедрения в образовательном учреждении. Особое внимание уделяется методическим аспектам использования чат-ботов как инструмента снижения бюрократической нагрузки на педагога. Статья адресована учителям, классным руководителям, педагогам-новаторам, методистам, руководителям образовательных организаций и специалистам в области цифровизации образования.

Ключевые слова: цифровизация образования, автоматизация, Telegram-бот, Python, парсинг данных, школьное питание, управление образовательным процессом.

Введение. Актуальность проблемы

Современная школа находится в стадии активной цифровой трансформации. Федеральные проекты и государственные стандарты

нацеливают образовательные учреждения на внедрение цифровых инструментов не только в учебный процесс, но и в систему управления. Однако, несмотря на наличие электронных журналов и дневников, значительная часть рутинной организационной работы учителя по-прежнему выполняется «в аналоговом режиме».

Одной из таких рутинных, но критически важных задач является организация питания обучающихся. Ежедневно классные руководители сталкиваются с необходимостью сбора информации о присутствующих учениках, выяснения их предпочтений в еде (завтрак, обед, льготное питание) и формирования поименной или количественной заявки для заведующего производством столовой.

Традиционный формат этой работы — бумажные формы обладают рядом существенных недостатков:

- **Высокие временные затраты.** Учитель тратит от 10 до 20 минут перемены на организационные моменты.
- **«Человеческий фактор».** Бумажные заявки теряются, почерк может быть неразборчив, при ручном подсчете итогового количества порций часто возникают арифметические ошибки.
- **Отсутствие оперативности.** Изменения в количестве учеников (кто-то заболел и ушел после первого урока) сложно оперативно передать в столовую.
- **Сложность отчетности.** Администрации школы и работникам столовой приходится вручную сводить данные от десятков классов в единую таблицу, что является крайне трудоемким процессом.

Таким образом, **актуальность** нашего проекта обусловлена необходимостью разрешения противоречия между возрастающими требованиями к эффективности работы педагога и сохранением устаревших, неэффективных механизмов административного взаимодействия.

Цель нашей работы заключалась в разработке, тестировании и внедрении автоматизированной системы сбора и обработки заявок на питание на базе мессенджера Telegram, позволяющей минимизировать участие человека в передаче данных и исключить ошибки.

Теоретическое обоснование и выбор инструментария

Чат-боты как элемент образовательной экосистемы

Приступая к разработке решения, мы проанализировали существующие подходы к автоматизации. Установка специализированного программного обеспечения (ПО) на компьютеры учителей часто невозможна из-за технических ограничений или требует длительного обучения персонала. Мобильные приложения требуют установки и занимают память устройства, что вызывает сопротивление пользователей.

В этом контексте использование технологии чат-ботов (Chat-bot) представляется наиболее перспективным. Согласно статистике, мессенджер Telegram установлен у подавляющего большинства педагогов. Следовательно, внедрение бота не требует:

- Установки дополнительного ПО.
- Регистрации новых аккаунтов.
- Длительного обучения (интерфейс мессенджера привычен).

С методической точки зрения, мы рассматриваем создание такого бота не просто как техническую задачу, но как пример **социально значимого проектирования**. Это демонстрирует, как прикладные навыки программирования могут решать реальные проблемы школьного сообщества.

Технологический стек проекта

Для реализации проекта нами был выбран язык программирования **Python**. Это решение обосновано несколькими факторами:

- **Популярность и поддержка.** Python обладает обширными библиотеками

для работы с API Telegram (в нашем случае использовалась асинхронная библиотека *python-telegram-bot*).

- **Обработка текста.** Python обладает мощнейшими встроенными инструментами для обработки строковых данных и регулярных выражений, что критически важно для реализованной нами функции извлечения текста (парсинга).
- **Интеграция с офисными приложениями.** Библиотека *openpyxl* позволяет напрямую генерировать отчеты в формате Excel (.xlsx), что было строгим требованием со стороны работников столовой.

Архитектура решения строится по принципу «клиент-сервер». База данных реализована на основе *SQLite* с последующей возможностью масштабирования.

Описание разработки и функционала системы

Мы разделили функционал бота на два ролевых уровня доступа: «**Классный руководитель**» и «**Администратор столовой**». В процессе проектирования интерфейса для учителей мы пришли к выводу, что ввод данных через множество кнопок может быть утомительным. Поэтому было принято нестандартное инженерное решение: гибридный ввод (кнопки + текстовый шаблон).

Алгоритм работы пользователя (Учителя)

Логика взаимодействия учителя с системой выстроена следующим образом:

1. **Авторизация.** При первом запуске бот сверяет ID пользователя (Telegram ID) с предварительно загруженным списком авторизованных педагогов. Это надежно защищает систему от спама и ложных заявок со стороны учеников.
2. **Выбор параметров.** Через кнопочное меню (Inline-кнопки) учитель выбирает дату подачи заявки («Сегодня» или «Завтра») и подтверждает свой класс.
3. **Формирование заявки (Текстовый ввод).** Для максимального упрощения

процесса мы реализовали подачу заявки обычным текстовым сообщением по простому шаблону. Учителю достаточно скопировать из своих заметок или напечатать список фамилий питающихся учеников, распределенных по категориям (например: "Завтрак: Иванов, Петров. Обед: Сидоров").

4. **Парсинг (синтаксический анализ) данных.** Это ключевой технический узел программы. Наш скрипт на Python автоматически обрабатывает входящее текстовое сообщение. Программный алгоритм извлекает фамилии обучающихся, распознает категории питания и автоматически производит математический подсчет порций.
5. **Подтверждение.** Бот присылает структурированное ответное сообщение для проверки: *«Ваша заявка: 5А класс. Завтраков (2): Иванов, Петров. Обедов (1): Сидоров. Итого: 3 порции. Верно?»*. После нажатия кнопки «Подтвердить», структурированные данные записываются в базу.

Важно отметить, что мы предусмотрели возможность редактирования. Если ситуация изменилась (ученик ушел домой), учитель просто отправляет новый список, и бот обновляет запись в базе данных за текущую дату.

Алгоритм работы Администратора (Столовой)

Для ответственного за питание функционал намеренно упрощен. Ему не нужно вводить данные, его цель — получение сводной информации.

1. **Панель управления.** У администратора в меню доступна кнопка «Сформировать отчет».
2. **Обработка данных.** При нажатии на кнопку скрипт обращается к базе данных, выбирает все подтвержденные списки на указанную дату, суммирует количество порций и группирует фамилии по классам.
3. **Генерация документа.** С помощью библиотеки *openpyxl* «на лету» создается файл Excel. Шаблон таблицы строго соответствует форме, принятой в столовой: в ней, присутствуют сводные цифры для поваров (сколько готовить) и детальные списки по классам для контроля раздачи.

4. **Отправка.** Бот присылает готовый файл .xlsx прямо в чат администратору за доли секунды.

Этапы внедрения и апробация

Внедрение проекта проходило в три этапа, что позволило избежать технического коллапса и негативной реакции коллектива.

Этап 1. Альфа-тестирование (техническое)

На этом этапе проверялась работа алгоритмов парсинга. Мы тестировали скрипт на сообщениях с опечатками, лишними пробелами и отсутствующими знаками препинания. Это позволило нам отладить код и доработать алгоритм так, чтобы программа корректно извлекала фамилии даже из неидеально отформатированного текста.

Этап 2. Пилотное внедрение (фокус-группа)

В тестировании приняли участие 5 классных руководителей из разных параллелей. В течение недели они дублировали заявки: сдавали привычный бумажный вариант и отправляли данные через бот.

Результаты пилота:

- Доработан алгоритм распознавания шаблонов сообщений (сделан более "гибким" к опечаткам).
- Скорректировано время автоматического закрытия приема заявок.
- Учителя отметили, что копирование готового списка фамилий в чат занимает считанные секунды.

Этап 3. Масштабирование

После финальной отладки система была представлена на педагогическом совете. Педагогам была выдана краткая инструкция-памятка по формату текстового сообщения.

Результаты и анализ эффективности

Спустя период опытно-промышленной эксплуатации мы провели анализ эффективности внедрения разработки.

Качественные показатели:

1. **Психологический комфорт.** Снизился уровень тревожности педагогов, связанный с необходимостью успеть физически отнести заявку по форме в столовую до определенного времени.
2. **Прозрачность и контроль.** В любой момент администрация видит точные поименные списки питающихся. Исключены ситуации потери данных.
3. **Разгрузка персонала столовой.** Заведующий производством больше не тратит 1-1,5 часа ежедневно на ручной подсчет порций и сведение тетради учета. Отчет формируется мгновенно.

Количественные показатели:

- Время на подачу заявки сократилось с **15 минут** (с учетом пути до столовой) до **одной минуты** (время на набор или вставку текста в мессенджер).
- Количество ошибок при подсчете порций снизилось до **0%** (математику выполняет сервер).
- Значительная экономия бумаги (порядка 1000 листов формата А4 за учебный год).

Заключение и перспективы развития

Мы разработали и успешно внедрили программный инструмент, который на практике доказал свою высокую эффективность. Проект является ярким примером того, как внутренняя инженерная разработка силами школы может решать насущные административные задачи лучше сторонних шаблонных систем.

Автоматизация рутинных процессов высвобождает главный ресурс учителя — время, которое теперь может быть направлено на непосредственную педагогическую деятельность, общение с детьми и профессиональное развитие.

Перспективы развития проекта:

1. **Интеграция с электронным журналом.** В перспективе возможна

настройка выгрузки списков отсутствующих из ЭЖД для автоматического их исключения из заявок на питание.

2. **Модуль финансового контроля.** Добавление функции подсчета баланса средств для классов, питающихся на платной основе.
3. **Аналитика дашбордов.** Внедрение модуля статистики для руководства школы, визуализирующего динамику охвата школьников горячим питанием в течение учебной четверти.

Список использованных источников:

1. Никулина Т. В., Стариченко Е. Б. Информатизация и цифровая трансформация образования: понятия и технологии [Электронный ресурс] // Инновационные проекты и программы в образовании. — 2018. — № 4. — С. 82–87. — <https://cyberleninka.ru/article/n/informatizatsiya-i-tsifrovizatsiya-obrazovaniya-ponyatiya-tehnologii-upravlenie> (дата обращения: 20.02.2026).
2. Официальная документация библиотеки python-telegram-bot [Электронный ресурс]. — Режим доступа: <https://docs.python-telegram-bot.org/> (дата обращения: 20.02.2026).
3. СанПиН 2.3/2.4.3590-20 «Санитарно-эпидемиологические требования к организации общественного питания населения» [Электронный ресурс]. — Режим доступа: <https://docs.cntd.ru/document/566276706> (дата обращения: 20.02.2026).
4. Свейгарт Э. Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих / Э. Свейгарт. — Москва: Вильямс, 2017. — 592 с.
5. Указ Президента Российской Федерации от 07.05.2024 г. № 309 «О национальных целях развития Российской Федерации на период до 2030 года и на перспективу до 2036 года» [Электронный ресурс]. — Режим доступа: <http://kremlin.ru/acts/bank/50542> (дата обращения: 20.02.2026).