

Программное обеспечение для автоматизации учета личных финансов на платформе Telegram: разработка и возможности применения

Гареев И.И.
Студент первого курса магистратуры,
бакалавр
Институт информатики, математики и
робототехники
Уфимский университет науки и технологий
Уфа, Россия
e-mail: gareev_ilyas@list.ru

Абдрахманова Р.П.
Доцент, к. ф-м. н.
Институт информатики, математики и
робототехники
Уфимский университет науки и технологий
Уфа, Россия
e-mail: Vmk_rimma@mail.ru

Аннотация¹

В настоящей работе обсуждаются результаты разработки программного обеспечения, предназначенного для автоматизации учета личных финансов.

Объектом исследования является процесс автоматизации учёта личных финансов с помощью чат-бота Telegram.

Предметом исследования является разработка чат-бота Telegram для автоматизации учёта личных финансов.

Разработано и реализовано программное обеспечение в виде чат-бота Telegram, позволяющего автоматизировать учёт личных финансов, управлять финансовыми целями.

Реализована возможность внесения чеков при помощи QR-кода, для автоматического заполнения информации о финансовой операции.

Реализована возможность анализа финансов с визуальным и текстовым описанием.

Ожидается, что предложенный программный способ учета позволит пользователю удобно управлять финансами.

Ключевые слова: учёт финансов; финансовые цели; анализ расходов; Telegram; чат-бот.

1. Введение

В современном мире, где технологии проникают в каждый аспект нашей жизни, автоматизация становится неотъемлемой частью повседневных

Труды X Международной научной конференции "Информационные технологии интеллектуальной поддержки принятия решений", 12-14 ноября, Уфа-Баку-Чандигарх, 2024

задач. Одной из таких задач является ведение учета личных финансов. Контроль над расходами и доходами важен для обеспечения финансовой стабильности и благосостояния. Однако учет вручную зачастую требует много времени и усилий, что делает его нецелесообразным для большинства людей.

Управление финансами помогает осознанно контролировать расходы и доходы. Информация за определенный период позволяет оценить финансовые возможности, чтобы избежать долгов и чувствовать себя уверенно.

Финансовые цели также играют важную роль. Постановка конкретных целей помогает сосредоточиться на том, чего мы хотим достичь в будущем и способствует принятию более взвешенных решений в отношении финансов.

Актуальность. Создание программного обеспечения для автоматизации учета личных финансов актуально в современном мире, где все больше людей сталкиваются с проблемой отслеживания своих расходов и доходов: они влияют на финансовую стабильность, позволяют планировать бюджет и достигать финансовых целей. Традиционные методы учета (электронные таблицы Excel, приложения для ведения бюджета и т. д.) могут быть неудобными и затратными времени, поэтому решение в виде чат-бота может значительно облегчить эту задачу.

Чат-боты предлагают удобный способ взаимодействия с программой финансового учета, позволяя вводить информацию о доходах и расходах и устанавливать финансовые цели прямо в чате, без необходимости заходить в специальное приложение или на веб-сайт. Такой подход может мотивировать людей более внимательно относиться к управлению финансами.

2. Публикация

Целью данной статьи является создание удобного и функционального инструмента для управления личными финансами посредством разработки чат-бота Telegram.

Исходя из анализа предметной области, можно сделать вывод, что существующие решения предлагают целый ряд полезных функций. Пользователям предоставляется удобный инструмент для контроля своих расходов и доходов, некоторые даже позволяют сканировать QR-коды с чеков. Тем не менее, во многих случаях доступ к полному набору функций ограничен и требует оформления платной подписки. Также стоит отметить отсутствие возможности устанавливать финансовые цели, что могло бы повысить эффективность управления финансами. В целом, предлагаемые решения представляют собой весьма функциональные, но не всегда гибкие и универсальные инструменты для ведения бюджета.

Проблема разработки программного обеспечения для автоматизации учёта личных финансов заключается в создании чат-бота, который бы упрощал процесс учёта финансов, обеспечивал удобную и доступную форму взаимодействия для пользователей, и в то же время обладал достаточным функционалом для правильного анализа и управления финансами.

Структура решения задачи:

1. Требования к функциональным возможностям чат-бота: изучение потребностей пользователей в учёте личных финансов, определение функциональных и нефункциональных требований к разрабатываемому чат-боту.
2. Архитектура и дизайн чат-бота: разработка структуры и архитектуры чат-бота, проектирование пользовательского интерфейса и функциональных возможностей.
3. Выбор технологий: выбор стека технологий для реализации чат-бота.
4. База данных: разработка базы данных, в которой будет храниться необходимая информация для контроля финансов.
5. Разработка и тестирование: программная реализация функций учёта финансов, тестирование функциональных возможностей на реальных сценариях использования.

Формальная постановка задачи

Контекстная диаграмма (рисунок 1) представляет собой схему автоматизации учета личных финансов.

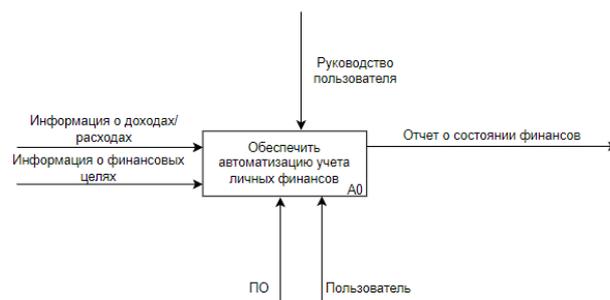


Рис. 1. Контекстная диаграмма ИС «Обеспечить автоматизацию учета личных финансов»

Входные данные:

1. Информация о доходах и расходах. Данная информация вводится вручную в виде числа или математического выражения, либо с помощью отправки фото QR-кода в чеке.
2. Информация о финансовых целях. Пользователь может добавить новую финансовую цель, редактировать и удалять существующие.

Результаты функционирования:

1. Отчет о состоянии финансов - анализ и категоризация доходов и расходов.

Механизм:

1. Программное обеспечение.
2. Пользователь.

Управление:

1. Руководство пользователя – инструкция по работе с приложением, predetermined разработчиком правила, в соответствии с которыми происходит взаимодействие с программой.

Далее на рисунке 2 представлена диаграмма декомпозиции, которая используется для разбиения и подробного описания основного процесса, представленного на контекстной диаграмме.

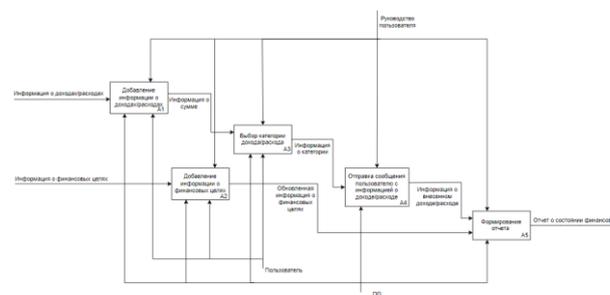


Рис. 2. Диаграмма декомпозиции «Обеспечить автоматизацию учета личных финансов»

Программное обеспечение для автоматизации учета личных финансов на платформе Telegram: разработка и возможности применения

Классификация подзадач. Для разработки чат-бота для автоматизации учёта личных финансов можно выделить следующие классы подзадач:

Управление финансами:

- Внесение данных о доходах и расходах;
- Редактирование и управление записями.

Управление финансовыми целями:

- Добавление новых целей;
- Редактирование существующих целей;
- Отслеживание прогресса достижения целей.

Аналитика и отчетность:

- Формирование отчетов;
- Визуализация данных.

Итак, каждая из этих задач требует внедрения в разработку чат-бота для автоматизации учёта личных финансов.

Рассмотрим *математические модели подзадач*, а также где они применяются:

- Модель управления финансами:

Данная модель предназначена для эффективного управления финансовыми операциями. Она предоставляет возможность добавлять, редактировать и удалять финансовые транзакции (доходы и расходы), предлагая пользователям инструменты для контроля своих финансов.

- Модель управления финансовыми целями:

Данная модель предназначена для эффективного управления финансовыми целями пользователя. Она предлагает функционал для добавления, редактирования и мониторинга финансовых целей, что способствует планированию расходов и достижению поставленных финансовых задач.

- Модель аналитики и отчетности:

Данная модель может использоваться для анализа финансов. Модель предоставляет инструменты для формирования отчетов о доходах и расходах пользователей за определенный период времени. Отчеты могут включать в себя суммарные показатели доходов и расходов, детализацию по категориям, а также визуализацию данных с помощью диаграмм для лучшего понимания финансового состояния.

Далее представлена математическая модель чат-бота для автоматизации учёта личных финансов на основе теории множеств.

Рассмотрим элементы, входящие в математическую модель:

Множество пользователей:

Пусть U – множество всех пользователей, которые имеют доступ к веб-интерфейсу. Каждый

пользователь $u \in U$ будет состоять из следующих атрибутов:

- Идентификатор пользователя: $id(u)$;
- Имя пользователя: $name(u)$.

Множество финансовых операций:

Пусть D — множество всех финансовых операций. Каждая финансовая операция $d \in D$ будет состоять из следующих атрибутов:

- Идентификатор финансовой операции: $id(d)$;
- Содержание финансовой операции: $content(d)$.

Множество финансовых целей:

Пусть G – множество всех финансовых целей. Каждая цель $g \in G$ будет состоять из следующих атрибутов:

- Идентификатор цели: $id(g)$;
- Содержание цели: $content(g)$;
- Статус цели: $status(g)$ (например, «текущая» или «завершенная»).

Множество отчетов:

Пусть R – множество всех отчетов. Каждый отчет $r \in R$ будет состоять из следующих атрибутов:

- Идентификатор пользователя: $id(r)$;
- Содержание отчета: $content(r)$.

Рассмотрим набор формул и операций, которые используют эти множества и их атрибуты для представления чат-бота:

Добавление нового пользователя:

$$U = U \cup \{u_{new}\},$$

где u_{new} — новый пользователь, добавляемый во множество пользователей U .

Добавление финансовой операции:

$$D = D \cup \{d_{new}\},$$

где d_{new} – новая финансовая операция, добавляемая во множество операций D .

Редактирование финансовой операции:

$$d_{edited} = D \setminus \{d_{old}\} \cup \{d_{new}\},$$

где d_{old} — первоначальная операция, d_{new} — отредактированная операция.

Удаление финансовой операции:

$$D = D \setminus \{d_{del}\},$$

где d_{del} — операция, которую нужно удалить из множества операций D .

Добавление финансовой цели:

$$G = G \cup \{g_{new}\},$$

где g_{new} – новая финансовая цель, добавляемая во множество целей D .

Редактирование финансовой цели:

$$g_{edited} = G \setminus \{g_{old}\} \cup \{g_{new}\},$$

где g_{old} — первоначальная цель, g_{new} — отредактированная цель.

Удаление финансовой цели:

$$G = G \setminus \{g_{del}\},$$

где g_{del} — цель, которую нужно удалить из множества задач G .

Формирование отчетов:

$$r(u) = \{d \in D(u) \mid date(d) \text{ in period}\},$$

где $r(u)$ — отчет о финансовых операциях пользователя $u \in U$, $date(d)$ — функция, возвращающая дату операции d , $date(d) \text{ in period}$ — это условие, указывающее, что дата операции d находится в пределах указанного периода времени.

Архитектура разрабатываемого программного продукта.

Чат-бот, предназначенный для автоматизации учета личных финансов, работает на основе клиент-серверной архитектуры.

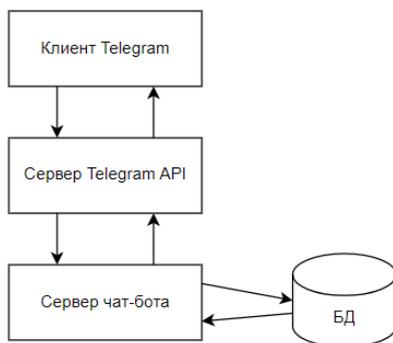


Рис. 3. Схема клиент-серверной архитектуры

На рисунке 3 показано взаимодействие между клиентом Telegram, сервером Telegram API, сервером чат-бота и базой данных.

Клиент (Telegram клиент) — это приложение Telegram, установленное на устройстве пользователя. Он предоставляет интерфейс для взаимодействия с пользователем и отправляет запросы к серверам Telegram API для получения и отправки сообщений.

Сервер API Telegram — это сервер, который предоставляет Telegram для обработки запросов от клиентов и возврата соответствующих ответов. Он осуществляет взаимодействие как с клиентами

Telegram, так и с сервером чат-бота, обеспечивая передачу сообщений и выполнение других функций.

Сервер чат-бота — это сервер, на котором размещен код чат-бота. Он принимает запросы от серверов Telegram API, обрабатывает их и возвращает соответствующие ответы. Такой сервер может быть развернут как на облачных платформах, так и на локальном компьютере.

База данных располагается на уровне сервера чат-бота и используется для хранения и извлечения данных, необходимых для его функционирования.

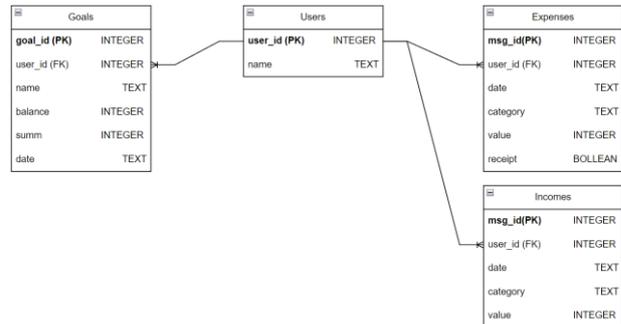


Рис. 4. Схема базы данных

Спроектированная для хранения данных пользователей база данных (рисунок 4) состоит из четырех таблиц:

1) Таблица Users. В данной таблице хранится информация о пользователях.

Каждая запись содержит следующие поля:

- **user_id**: уникальный идентификатор пользователя в Telegram. Этот параметр
- выступает в качестве первичного ключа и используется для однозначной идентификации пользователя;
- **name**: имя пользователя. Это текстовое поле, которое содержит имя

пользователя в Telegram.

2) Таблица Goals. В данной таблице хранится информация о финансовых целях

пользователей. Каждая запись в таблице содержит следующие поля:

- **goal_id**: уникальный идентификатор цели. Это первичный ключ, который используется для однозначной идентификации цели;
- **user_id**: идентификатор пользователя, который создал цель. Это внешний ключ, связывающий цель с определенным пользователем из таблицы Users;
- **name**: название цели. Это текстовое поле, содержащее описание финансовой цели;
- **balance**: текущая сумма накоплений по цели. Это числовое поле, указывающее на текущий прогресс пользователя в достижении цели;

- `summ`: конечная сумма цели. Это числовое поле, определяющее сумму, которую пользователь стремится накопить;
- `date`: дата достижения цели. Это текстовое поле, указывающее на дату, когда
- пользователь планирует достичь своей финансовой цели.

3) Таблица `Expenses`. В данной таблице хранится информация о расходах пользователей. Каждая запись в таблице содержит следующие поля:

- `msg_id`: идентификатор сообщения, через которое был добавлен расход. Этот параметр помогает связать расход с конкретным сообщением в мессенджере;
- `user_id`: идентификатор пользователя, который совершил расход. Это внешний ключ, связывающий расход с определенным пользователем из таблицы `Users`;
- `date`: это текстовое поле, указывающее на дату, когда был совершен расход;
- `category`: это текстовое поле, которое содержит категорию расхода;
- `value`: это числовое поле, которое содержит сумму расхода;
- `receipt`: это логическое поле, указывающее, является ли расход чеком.

4) Таблица `Incomes`. В данной таблице хранится информация о доходах пользователей. Каждая запись в таблице содержит следующие поля:

- `msg_id`: идентификатор сообщения, через которое был добавлен доход;
- `user_id`: идентификатор пользователя, который получил доход. Это внешний ключ, связывающий доход с определенным пользователем из таблицы `Users`;
- `date`: это текстовое поле, указывающее на дату, когда был совершен расход;
- `category`: это текстовое поле, которое содержит категорию расхода;
- `value`: это числовое поле, которое содержит сумму расхода.

Все эти таблицы взаимосвязаны друг с другом и обеспечивают хранение и управление финансовыми данными пользователей в рамках чат-бота.

Алгоритм работы программного обеспечения. Описание алгоритма работы чат-бота для автоматизации учёта личных финансов:

1. Запуск бота:
 - Загрузка ресурсов;
 - Чтение конфигурационных файлов и базы данных;
 - Нажать кнопку «Старт».
2. Пользователь выбирает действие:
 - Добавить доход/расход;
 - Добавить цель;
 - Добавить средства к цели;

- Просмотреть отчет.
3. Если пользователь выбирает «Добавить доход/расход», то происходят следующие действия:
 - Если пользователь выбирает «Добавить сумму вручную», то ввод суммы, иначе — отправка фото чека с QR-кодом;
 - Выбор категории;
 - Данные сохраняются в базе данных.
 4. Если пользователь выбирает «Добавить цель», то происходит следующее действие:
 - Заполнение информации о цели;
 - Данные сохраняются в базе данных.
 5. Если пользователь выбирает «Добавить средства к цели», то происходят следующие действия:
 - Если у пользователя есть цели, то ввод средств, выбор цели;
 - Данные сохраняются в базе данных.
 6. Если пользователь выбирает «Просмотреть отчет», то происходит следующее действие:
 - Вывод отчета доходов/расходов по месяцам.
 7. Завершение работы с чат-ботом:
 - Закрытие чата с чат-ботом.

Программное обеспечение.

Язык программирования. Для создания Telegram бота был выбран язык программирования Python. Он предоставляет оптимальный баланс между удобством использования, функциональностью и производительностью. Простота Python и его богатая экосистема библиотек делают его отличным выбором для разработки чат-ботов. Кроме того, Python обладает большим и активным сообществом, что гарантирует поддержку и доступ к множеству различных ресурсов и инструментов для разработки.

Фреймворк для взаимодействия с Telegram API. Aiohttp выделяется асинхронной архитектурой, удобным API и широкими возможностями, что делает его оптимальным выбором для разработки чат-ботов. Кроме того, Aiohttp поддерживается активным сообществом, что обеспечивает надежность и долгосрочную поддержку.

Система управления базами данных (СУБД). SQLite отличается простотой в использовании, не требует настройки сервера и обеспечивает высокую производительность для небольших проектов. Это позволяет разработчикам сосредоточиться на реализации функциональности бота, минимизируя усилия на управление базой данных.

Обсуждение результатов. В ходе проделанной работы был разработан чат-бот, в котором имеется возможность выполнения перечисленных ниже функций:

- Создание пользователем новой записи о финансовой операции.
- Возможность выбора категории дохода или расхода.

- Обработка изображений чеков для автоматического заполнения информации.
- Управление финансовыми целями (добавление, редактирование, удаление)
- Сохранение данных о финансовой операции в базе данных.
- Просмотр детальной информации, редактирование и удаление определенной записи.
- Автоматическое формирование отчетов о доходах и расходах.
- Просмотр графических и текстовых представлений анализа доходов и расходов.
- Просмотр курса валют и возможность конвертации.
- Получение полезных советов по ведению бюджета.



Рис. 5. Экранные формы разработанного чат-бота

На рис. 5 представлены экранные формы чат-бота.

3. Заключение

В ходе работы было разработано программное обеспечение в виде чат-бота Telegram для автоматизации и удобства учета личных финансов. Он позволяет добавлять информацию о доходах и расходах по категориям, устанавливать и отслеживать финансовые цели, а также генерировать отчеты о расходах и доходах по месяцам.

Список используемых источников

1. 19 самых удобных программ для ведения бюджета [Электронный ресурс]. – URL: <https://lifelhacker.ru/family-budget/> (дата обращения: 10.02.2024).
2. Matplotlib [Электронный ресурс]. – URL: <https://matplotlib.org/stable/index.html> (дата обращения: 15.04.2024).
3. Python Requests Library [Электронный ресурс]. – URL: <https://realpython.com/python-requests/> (дата обращения: 29.03.2024).

4. Pyzbar [Электронный ресурс]. – URL: <https://pypi.org/project/pyzbar/> (дата обращения: 11.04.2024).
5. StackOverflow [Электронный ресурс]. – URL: <https://ru.stackoverflow.com/> (дата обращения: 09.04.2024).
6. Telegram Bot API [Электронный ресурс]. – URL: <https://core.telegram.org/bots/api> (дата обращения: 15.03.2024).
7. Wikipedia – свободная энциклопедия [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/Финансы> (дата обращения: 20.02.2024).
8. Документация Aiogram [Электронный ресурс]. – URL: <https://docs.aiogram.dev/en/latest/> (дата обращения: 0 9.0 4.20 24).
9. Документация DBEaver [Электронный ресурс]. – URL: <https://dbeaver.com/docs/dbeaver/> (дата обращения: 0 5.0 4.20 24).
10. Документация PyCharm [Электронный ресурс]. – URL: <https://www.jetbrains.com/help/pycharm/getting-started.html> (дата обращения: 09.03.2024).
11. Документация Python [Электронный ресурс]. – URL: <https://docs.python.org/3/> (дата обращения: 13.03.2024).
12. Документация SQLite [Электронный ресурс]. – URL: <https://www.sqlite.org/docs.html> (дата обращения: 17.0 3.20 24).

13. Златопольский, Д. М. Основы программирования на языке Python. / Д. М. Златопольский. – Москва: МК Пресс, 2017. – 284 с.
14. Измерение уровня финансовой грамотности [Электронный ресурс]. – URL: http://www.cbr.ru/analytics/szpp/fin_literacy/fin_ed_intro/ (дата обращения: 20.02.2024).
15. Клейсон Дж. С. Самый богатый человек в Вавилоне / Дж. С. Клейсон. – Москва: Попурри, 2020. – 160 с.
16. Клиент-серверная архитектура [Электронный ресурс]. – URL: <https://habr.com/ru/articles/495698/> (дата обращения: 20.03.2024).
17. Математические основы моделирования и алгоритмизации сложных систем. [Электронный ресурс]. – URL: <https://dit.isuct.ru/IVT/BOOKS/Model/Model2/part2.html> (дата обращения: 29.02.2024)
18. Тарасова, А Сам себе финансист / А Тарасова. – Москва: Альпина Паблишер, 2018. – 192 с.
19. Телеграм-боты на Python и Aiogram [Электронный ресурс]. – URL: <https://stepik.org/course/120924/syllabus> (дата обращения: 01.03.2024).
20. Хилл, Н. Думай и богатей / Н. Хилл. – Санкт Петербург: Прайм – ЕВРОЗНАК, 2022. – 384 с.

Программное обеспечение для автоматизации учета личных финансов на платформе Telegram: разработка и возможности применения