ОБРАЗОВАНИЕ

Кейс от Т-Банка

Рациональный ассистент

Описание кейса

Все мы сталкивались хоть раз с таким явлением, как импульсивные покупки, когда сиюминутное желание приобрести ту или иную вещь брало вверх над всеми другими потребностями.

Импульсивные покупки — не просто "слабость", а реальная угроза финансовому благополучию.

Каждый год миллионы людей тратят деньги на вещи, которые не используются и быстро забываются. Часто такие траты связаны с эмоциональным состоянием: стресс, скука, усталость — и вдруг "лечится" новым гаджетом, одеждой или дорогим хобби.

Результат?

* Пустой кошелёк;
* Разочарование;
* Нарушенные финансовые планы.

Важно показать покупателю, что его желание импульсивных покупок размывает границы умеренного потребления. В современном мире есть множество подходов, которые позволяют остудить желание потратить деньги "здесь и сейчас", а так же оценить последствия таких трат.

Создайте цифрового ассистента, который поможет пользователю остыть перед покупкой и принять осознанное решение.

Ваш продукт должен:

* Анализировать желаемую покупку;
* Учитывать цену, категорию, финансовые возможности пользователя;
* Применять период "охлаждения" — задержку перед покупкой;
* Блокировать запрещённые категории (игры, техника, шопинг и др.);
* Рассчитывать, когда пользователь сможет позволить себе покупку, исходя из накоплений;
* Хранить "желаемое" в личном кабинете и регулярно опрашивать пользователя: «Ты всё ещё хочешь это?».

Техническое задание

Базовый сценарий использования приложения

Чтобы создать наилучшие условия для благоприятного финансового положения пользователя, предлагается реализовать продукт, который позволит организовать процесс "охлаждения" покупок.

Пример решения:

Пользователь настраивает ИС по следующим параметрам:

- Запрещенные категории - если пользователь хочет запретить себе покупать вещи из определенной категории (бытовая техника, видеоигры, азартные игры и т.д.), то он может сохранить список этих категорий, чтобы ИС напоминала о самозапрете;
- Диапазоны охлаждения - в зависимости от суммы покупки, пользователь может указать разные сроки охлаждения покупки. Например - если покупка заберет у пользователя от 80.000 рублей, то тогда срок охлаждения составит 30 дней, от 150.000 рублей - 90 дней и так далее;
- Сумма накоплений - если пользователь хочет рассчитывать срок охлаждения с учетом фин. достатка, то он может включить функцию "Учитывать текущие накопления" в настройках. Тогда ему нужно указать, какую сумму в день/неделю/месяц он откладывает;
- Пользователь присылает ссылку на товар, который он хочет купить / присылает описание товара;
- ИС сканирует ссылку на товар, ищет там:

Название;
Цену;
Категорию.

Техническое задание

Если пользователь включил функцию "Учитывать текущие накопления" в настройках, то:

- ИС просит пользователя прислать размер текущих накоплений;
- Если присланная пользователем сумма меньше стоимости покупки, то ИС должна рассчитать, через сколько дней пользователь, без вреда для своего финансового положения, сможет совершить покупку;
- ИС анализирует полученные данные по следующим критериям:
- Цена - если сумма находится в диапазоне A, то предлагает подумать над покупкой N дней. Следовательно, если цена покупки будет находится в диапазоне B, C и т.д., то будет предложено отложить покупку на K, I и т.д. дней. Если срок рассчитывался ранее через функцию "Учитывать текущие накопления", то тогда ИС должна рассчитать общий рекомендованный срок;
- Категория - если категория находится в запрещенных, то ИС должна порекомендовать пользователю не совершать эту покупку и закончить сценарий коммуникации;
- После общих расчетов, ИС добавляет товар в личный кабинет пользователя. После добавления товара в ЛК, ИС должна раз в N-ое время нотифицировать пользователя опросом о необходимости покупки того или иного товара. Если товаров в ЛК несколько, ИС должна учесть это и должна составить общий опрос по всем сохраненным товарам.

Требования по платформе - мобильное приложение, веб приложение с адаптацией под ПК и мобильные устройства.

Ограничения и советы

* Авторизация/регистрация в приложении не нужны (!!!);
* В случае реализации чат-бота в Telegram вместо полноценного решения, будет оценён только сырой бизнес-функционал;
* Сначала реализуйте базовую логику, эвристики и общую связность бэкенда с фронтендом, затем переходите к более сложным требованиям;
* Макет в Figma/любом другом инструменте прототипирования или дизайна - не является решением и не будет оценен;
* Если предполагается реализация Web-приложения - рекомендуется использовать UI-Kit для работы с готовыми компонентами.

Объем по MoSCoW

Must (обязательно)

Настройка "контекста" пользователя, личный финансовый профайл - сколько откладывает в месяц, какой уровень ЗП;
Формат ввода данных "о себе" - анкета;
Ввод данных о желаемой покупке:

* Ввод названия покупки (название товара);
* Ввод цены товара;
* Ввод категории;

Настройка нотификатора:

* Как часто опрашивать о запланированных покупках;
* Какие товары нужно исключить из нотификации;
* Канал нотификации (уведомления, Telegram, почта (SMTP) и т.д.);

Нотификации должны перенаправлять пользователя в интерфейс приложения для "инвентаризации" желаемых покупок.

Must (обязательно)

Эвристики:

Диапазоны охлаждений - возможность ввести диапазоны объемов трат, при которых будут применяться разные правила "охлаждения" покупки, например:

* до 15.000 рублей - сутки;
* от 15.000 до 50.000 - неделя;
* от 50.000 до 100.000 рублей - месяц;
* и т.д. - потолка у диапазонов нет;

Период накопления - возможность оценить, через сколько можно будет совершить покупку при условии, что за покупку не будут отданы последние деньги, например:

* если у меня сейчас на счетах есть свободные 80.000 рублей, я откладываю по 5.000 рублей в месяц на свободные траты, а покупка стоит 100.000 рублей, то приложение должно посчитать, через сколько эта покупка станет комфортной (например, после этой покупки у меня останется как минимум половина от всего объема трат);

Если у решения есть Backend приложение - в нем должен быть рабочий Swagger (или любая другая автоматическая документация на основе OpenAPI приложения).

Should (желательно)

Рабочая система нотификации:
Умеет отправлять уведомления хотя бы по одному каналу нотификации;
Имеет готовый набор шаблонов времени нотификации, например:

* Раз в неделю;
* Раз в день;
* Раз в месяц;

Полноценный личный кабинет:

* Вход по никнейму (без авторизации!!!!);
* Вывод всех желаемых покупок, история отмененных или совершенных покупок;
* Финансовый профайл.

Could (по возможности)

Blacklist категорий для покупки:
Возможность ввести/выбрать из списка категории товаров, на которые деньги точно тратить нельзя;
ML/AI компонент, который сможет соотнести категорию товара с запрещенными категориями, например:

* У товара категория "Игровая консоль", в черном списке есть категория "техника" - компонент должен соотнести две эти категории и дать предположение, насколько они являются родственными (скоринг синонимов);

Если это будет интеграция в LLM, то необходимо предоставить возможность менять промт запроса через отдельный интерфейс приложения (аля админ панель с одной функцией);
Альтернативный формат ввода данных "о себе" - чат бот в интерфейсе приложения, который спарсит из ответ нужные данные и сам составит анкету пользователя;
Альтернативный формат ввода данных о товаре - возможность прислать ссылку на товар для дальнейшего парсинга указанной страницы.

Won’t (не желательно, не приветствуется)

Авторизация/регистрация - не нужна. Либо в приложении вообще нет аутентификации, либо она ограничена вводом никнейма/номера телефона и т.д.;

Интерфейс в чат-боте (например, решение в виде Telegram-бота).

ОБРАЗОВАНИЕ