Business Analyst

Компания создает MVP системы для выдачи годовых займов потенциальным клиентам.

**Процесс следующий:**

а) На одной страничке собираем у клиента личные данные, затем анализируем их и принимаем решение – выдавать клиенту займ на 1 год или нет.

б) Решение показываем клиенту в виде другой странички, где пишем либо “Простите, мы не можем вам помочь” или “Ваш займ готов, мы готовы выдать вам X тенге на 1 год. Вернуть нужно будет Y тенге! Для получения денег, пожалуйста, позвоните по телефону ...” Также решение должно посылаться клиенту на email.

* От клиента нам нужно собирать его имя, дату рождения, email, телефон, место работы (в виде списка основных индустрий и отдельного пункта для отсутствия занятости), месячную зарплату, наличие других задолженностей (если есть – то сколько платит по другим кредитам в месяц). Данные каждого нового клиента мы сохраняем в базу данных.

Анализ данных таков:

● если клиенту меньше 18 или больше 90 – то в займе будет отказано

● если клиент зарабатывает меньше 50000 тенге в месяц – отказ X тенге

● если клиент нигде не работает – отказ

Расчет кредитного предложения:

● если клиенту не был дан отказ, мы рассчитываем сумму займа по формуле (зарплата – месячные выплаты по другим задолженностям)\*10 и сумму для возврата с учетом того, что процентная ставка составляет 3% в месяц от суммы основного долга.

Дополнительно необходимо, чтобы телефон в успешном сообщении можно было настраивать через переменную в базе данных.

Задание:

● написать спецификацию с описанием UI, черновыми мокапами, функциональными требованиями к бэк-енд логике и расчету сумм (с примером)

● создать диаграмму процесса с использованием общеприменимой нотации

Дополнительное задание:

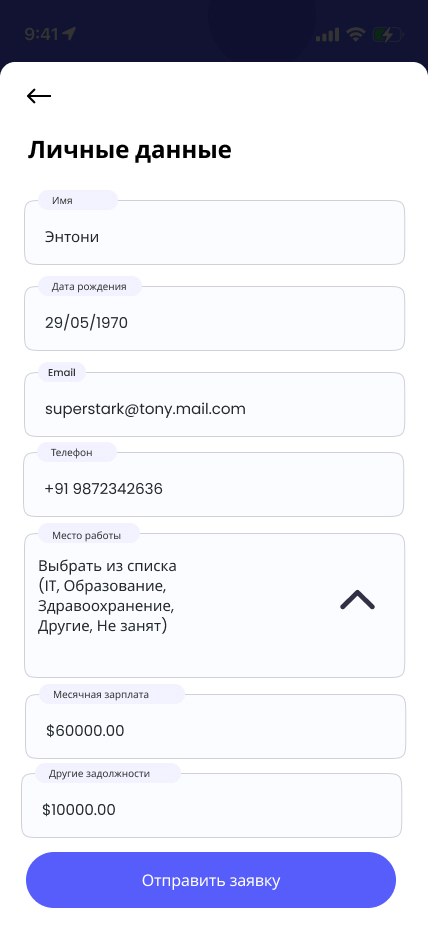
● в свободной форме предложить идеи по улучшению процесса. \*

После выполнения задания отправьте, пожалуйста, письмо на laba@solvatech.kz с выполненным ТЗ (ссылка на GitHub), а также вашим резюме. Мы оценим вашу работу в течение двух рабочих дней и вернемся с обратной связью в случае успешного выполнения задания.

#### **Описание интерфейса пользователя (UI)**

##### **Страница 1: Ввод данных клиента**

* **Форма для заполнения:**
  1. Поле для имени клиента (текстовое поле).
  2. Поле для даты рождения (дата).
  3. Поле для email (текстовое поле с валидацией email).
  4. Поле для телефона (текстовое поле с валидацией формата номера).
  5. Выпадающий список для выбора места работы с опцией "Не занято".
  6. Поле для указания месячной зарплаты (числовое поле).
  7. Поле для указания других задолженностей (числовое поле).
* **Кнопка**: "Отправить заявку".

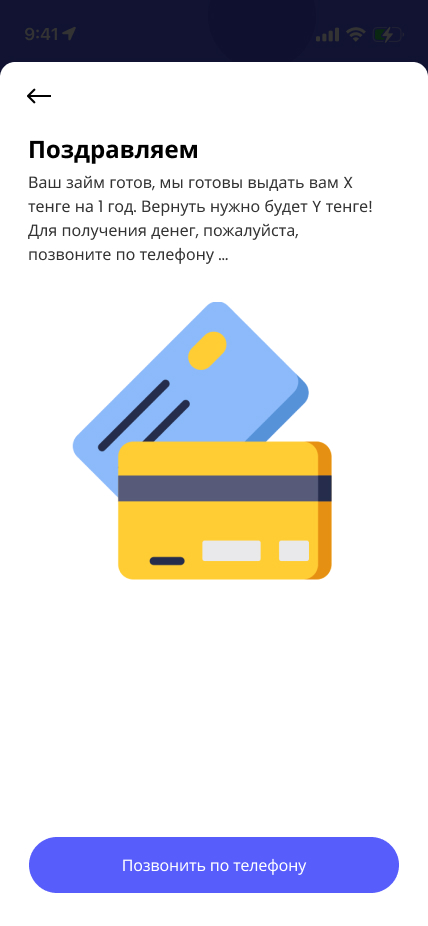


##### **Страница 2: Результат заявки (UI)**

* **Сообщение об отказе**:
  + "Простите, мы не можем вам помочь."



* **Сообщение об успешной выдаче займа**:
  + "Ваш займ готов, мы готовы выдать вам X тенге на 1 год. Вернуть нужно будет Y тенге! Для получения денег, пожалуйста, позвоните по телефону ..."



### **Функциональные требования к бэкенду**

#### **1. Получение данных от клиента**

* При получении данных от клиента:
  + Проверить корректность данных (валидация полей):
    - **Имя**: только буквы, длина не менее 2 символов.
    - **Дата рождения**: формат YYYY-MM-DD, не позже текущей даты.
    - **Email**: проверка на соответствие формату name@domain.com.
    - **Телефон**: только цифры, длина 10–15 символов.
    - **Место работы**: одно из предустановленных значений или "Не занят".
    - **Зарплата и задолженности**: только положительные числа.
  + Сохранить все данные в базе данных после успешной валидации.
  + Обеспечить логирование ошибок валидации для дальнейшего анализа.

#### **2. Анализ данных**

* Проверки возраста клиента:
  + Рассчитать возраст на основе даты рождения.
  + Если возраст < 18 или > 90 лет, вернуть сообщение об отказе.
* Проверка зарплаты:
  + Если зарплата < 50000 тенге, вернуть сообщение об отказе.
* Проверка занятости:
  + Если клиент не работает, вернуть сообщение об отказе.

#### **3. Расчет займа**

* + Ограничить минимальную сумму займа, например, 100000 тенге, и максимальную, например, 5000000 тенге.
  + Если итоговая сумма займа выходит за границы, вернуть сообщение об отказе.

Если все проверки пройдены:

* **Рассчитать сумму займа:** сумма\_займа = (зарплата - задолженности) \* 10

Пример:

Зарплата: 100 000 тенге

Задолженности: 0 тенге

Сумма займа: сумма\_займа = (100 000 - 0) \* 10 = 1 000 000 тенге

* **Рассчитать сумму для возврата с учетом процентной ставки:**  
  сумма\_возврата = сумма\_займа + (сумма\_займа \* 0.03 \* 12)

Сумма возврата: сумма\_возврата = 1 000 000 + (1 000 000 \* 0.36) = 1 360 000 тенге

Ежемесячный платеж: ежемесячный\_платеж = 1 360 000 / 12 ≈ 113 333 тенге

Процентная ставка в месяц составляет 3%.

Годовая ставка = 3% \* 12 = 36%.

#### **4. Формирование ответа**

* Если клиенту отказано: Возвратить сообщение: Простите, мы не можем вам помочь.
  + Сохранить отказ с причиной в базе данных.
* Если займ одобрен: Сформировать сообщение: Ваш займ готов, мы готовы выдать вам X тенге на 1 год. Вернуть нужно будет Y тенге! Для получения денег, пожалуйста, позвоните по телефону Z.
  + Сохранить информацию о расчете (сумма займа, сумма возврата) в базе данных.
  + Сохранить статус заявки как "Одобрено".

#### **5. Отправка уведомлений**

* **Email**:
  + Отправить клиенту уведомление о результате заявки:
    - Для отказа: причина отказа.
    - Для одобрения: детали займа (сумма займа, сумма возврата, телефон для связи).
  + Убедиться, что email отправлен, и записать статус отправки в базу данных.
  + При сбое отправки уведомления повторить попытку 3 раза с интервалом в 5 минут.
* **Логирование**:
  + Логировать все успешные и неуспешные попытки отправки email.

#### **6. Конфигурации:**

* Номер телефона в успешном сообщении:
  + Получать из таблицы settings в базе данных.
  + Предусмотреть возможность обновления номера через административный интерфейс или SQL-запрос.
  + Проверять корректность формата при обновлении.

#### **7. Хранение данных**

* Таблица заявок:
  + Сохранение:
    - Данные клиента.
    - Результат анализа (одобрено/отказано).
    - Причину отказа или детали займа.
    - Время создания и обновления записи.
* Таблица уведомлений:
  + Сохранение:
    - Тип уведомления (email, SMS и т. д.).
    - Статус отправки (успешно/ошибка).
    - Время отправки.

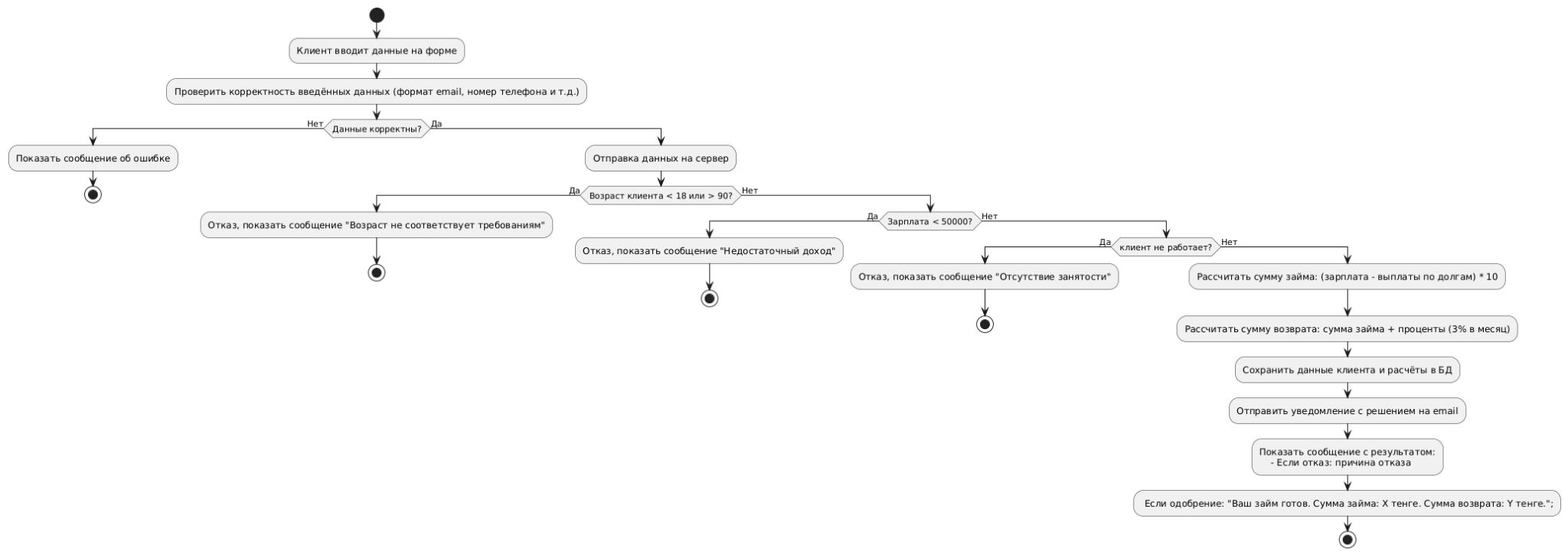
#### **8. Интеграция с внешними сервисами**

* Проработать интеграции с:
  + Сервисами проверки кредитной истории.
  + Сервисами отправки SMS-уведомлений (опционально).
* Обеспечить обработку ошибок при взаимодействии с внешними системами.

#### **9. Тестирование**

* Проработать автоматические тесты.

#### **Sequence-диаграмма.**



### **Дополнительное задание.**

**Интеграция с кредитными бюро**:

* Проверка кредитной истории клиента для повышения точности решений.

**Поддержка нескольких языков**:

* Добавление локализации для повышения удобства пользователей.

**Отображение деталей расчета**:

* Показывать клиенту, как именно была рассчитана сумма займа(кредитный калькулятор).