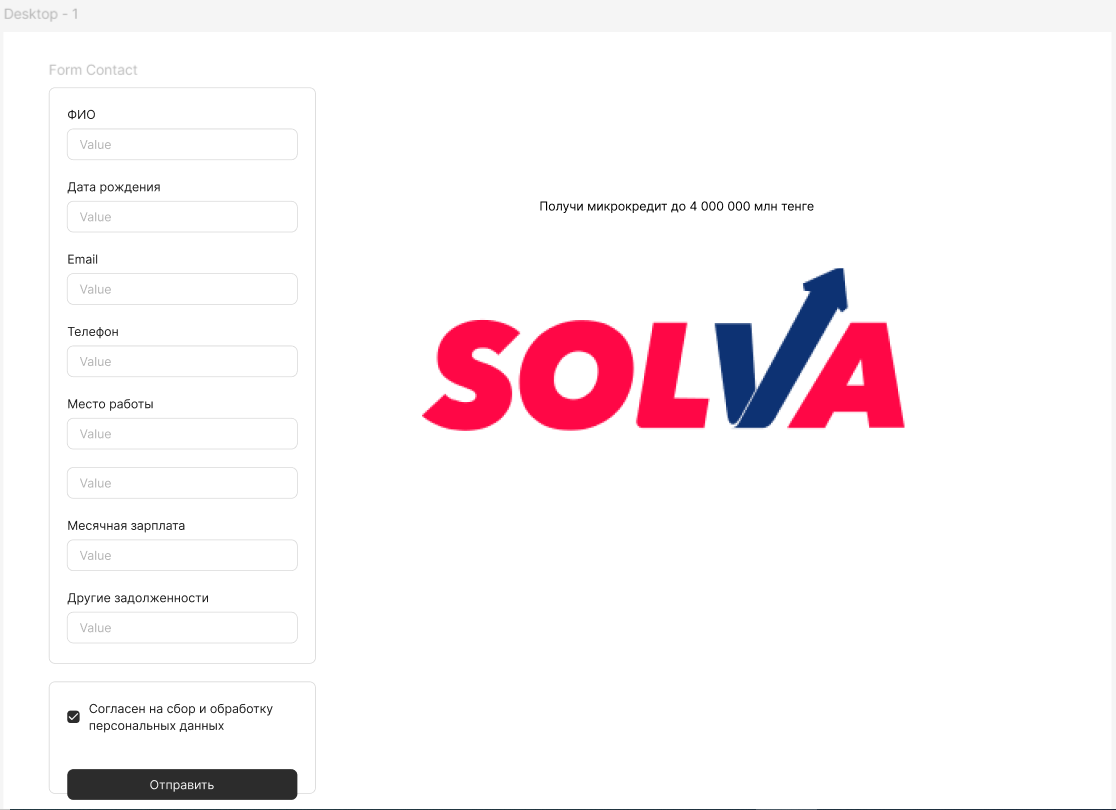
### Спецификация UI (детальное):

#### Страница 1: Ввод данных клиента

* **Имя клиента**:
  + Текстовое поле.
  + Валидация: обязательное поле.
* **Дата рождения**:
  + Поле выбора даты.
  + Валидация: обязательное поле, проверка возраста (18-60 лет).
* **Email**:
  + Текстовое поле для ввода.
  + Валидация: обязательное поле, проверка корректного формата email.
* **Телефон**:
  + Текстовое поле с маской для ввода номера телефона.
  + Валидация: обязательное поле, формат номера (например, +7 ХХХ ХХХ ХХХХ).
* **Место работы**:
  + Текстовое поле
  + Выпадающий список с основными индустриями:
    - IT
    - Финансы
    - Образование
    - Строительство
    - Без работы
  + Валидация: обязательное поле.
* **Месячная зарплата**:
  + Числовое поле.
  + Валидация: обязательное поле, минимум 85 000 тенге.
* **Другие задолженности**:
  + Числовое поле для ежемесячных выплат по другим кредитам.
  + Валидация: можно оставлять пустым, либо вводить положительное значение.
* **Согласие на обработку персональных данных**:
  + Чекбокс.
  + Валидация: обязательное поле.
* **Кнопка отправки**:
  + "Отправить данные" — отправляет информацию на сервер для обработки.

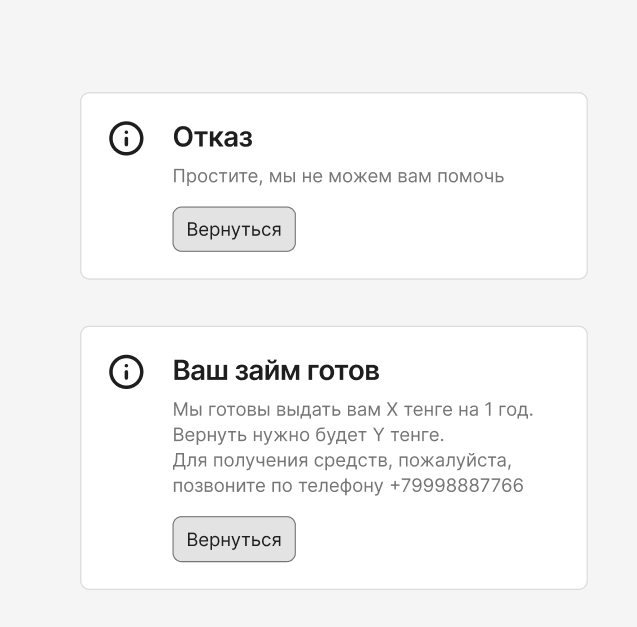
#### Пример чернового мокапа для первой страницы:



#### Страница 2: Результат анализа

1. **Отказ:**
   * Сообщение: "Простите, мы не можем вам помочь."
2. **Одобрение:**
   * Сообщение: "Ваш займ готов. Мы готовы выдать вам X тенге на 1 год. Вернуть нужно будет Y тенге!"
   * Номер телефона для связи, который должен настраиваться через базу данных.

#### Пример чернового мокапа для второй страницы:



**Функциональные требования к бэкэнду и логика расчетов займа**

### ****1.Модуль валидации данных клиента****

#### Требования:

* Backend должен принимать данные клиента, отправленные с фронтенда (имя, дата рождения, email, телефон, место работы, месячная зарплата, задолженности и выплаты по ним).
* Все данные должны проходить базовую валидацию на стороне сервера:
  + **Имя**: строка не пустая, не содержит цифр.
  + **Дата рождения**: проверка возраста (должно быть от 18 до 60 лет включительно).
  + **Email**: соответствует формату email (регулярное выражение).
  + **Телефон**: соответствует шаблону номера (регулярное выражение).
  + **Место работы**: допустимое значение из списка или опция "Без работы".
  + **Месячная зарплата**: целое число больше 85000.
  + **Другие задолженности**: целое число, если задолженности имеются, поле может быть пустым, если нет задолженностей.
  + **Согласие на обработку персональных данных**: обязательное поле (чекбокс должен быть отмечен).

#### Обработка:

* Если данные некорректны, backend возвращает ошибку с описанием причины (например, "Зарплата слишком низкая", "Возраст вне допустимого диапазона").

### 2. ****Модуль анализа данных****

#### Логика:

* После успешной валидации backend должен проверить следующие критерии:
  + **Возраст**: клиенту должно быть от 18 до 60 лет.
  + **Зарплата**: клиент должен зарабатывать больше 84999 тенге в месяц.
  + **Место работы**: клиент должен быть занятым, если выбрано "Без работы", отказ.
  + **Задолженности**: если задолженности указаны, суммы выплат должны быть больше нуля.

#### Обработка:

* Если хотя бы одно условие не выполнено, возвращается отказ с указанием сообщения: "Простите, мы не можем вам помочь."

### 3. ****Модуль расчета суммы займа****

#### Логика:

* Если клиент прошел все проверки, backend должен выполнить расчет возможной суммы займа:
  + Формула расчета суммы займа: (зарплата - выплаты по другим задолженностям) \* 10.
  + Процентная ставка: 3% в месяц.
  + Сумма для возврата: начисляется 3% на основной долг каждый месяц.

и вернуть сообщение:

* + Сообщение: "Ваш займ готов. Мы готовы выдать вам (расчетная сумма займа) тенге на 1 год. Вернуть нужно будет (расчетная сумма для возврата) тенге!"
  + Номер телефона для связи

#### Пример расчета:

* Зарплата: 50,000 тенге.
* Выплаты по задолженностям: 1,000 тенге.
* Сумма займа: (50000 - 1000) \* 10 = 490,000 тенге.
* Сумма к возврату (через 12 месяцев):  
  490,000 \* (1 + 0.36) = 666,400 тенге.

### 4. ****Модуль сохранения данных клиента****

#### Требования:

* Все данные клиента, включая решение по займу, должны быть сохранены в базе данных:
  + Имя, дата рождения, email, телефон.
  + Место работы, зарплата, задолженности.
  + Сумма займа, сумма к возврату, решение (одобрено/отказано), дата принятия решения.

### 5. ****Модуль отправки решения по email****

#### Логика:

* Backend должен отправить клиенту email с результатом анализа:
  + **Тема письма**: "Ваш займ: решение по заявке".
  + **Тело письма**: если одобрено, указать сумму займа, сумму к возврату и телефон для связи в сообщении: "Ваш займ готов. Мы готовы выдать вам (сумма займа) тенге на 1 год. Вернуть нужно будет (сумма к возврату) тенге!"

Если отказ, указать текст: "Простите, мы не можем вам помочь."

Пример email (одобрено):

Тема: Ваш займ: решение по заявке

Уважаемый Иван Иванов! Ваш займ готов. Мы готовы выдать вам 490,000 тенге на 1 год. Вернуть нужно будет 666400 тенге! Для получения средств, пожалуйста, позвоните по телефону: +79998887766

Пример email (отказ):

Тема: Ваш займ: решение по заявке

Уважаемый Иван Иванов! К сожалению, мы не можем вам помочь. Спасибо за обращение!

### 6. ****Модуль настройки телефона для связи****

#### Требования:

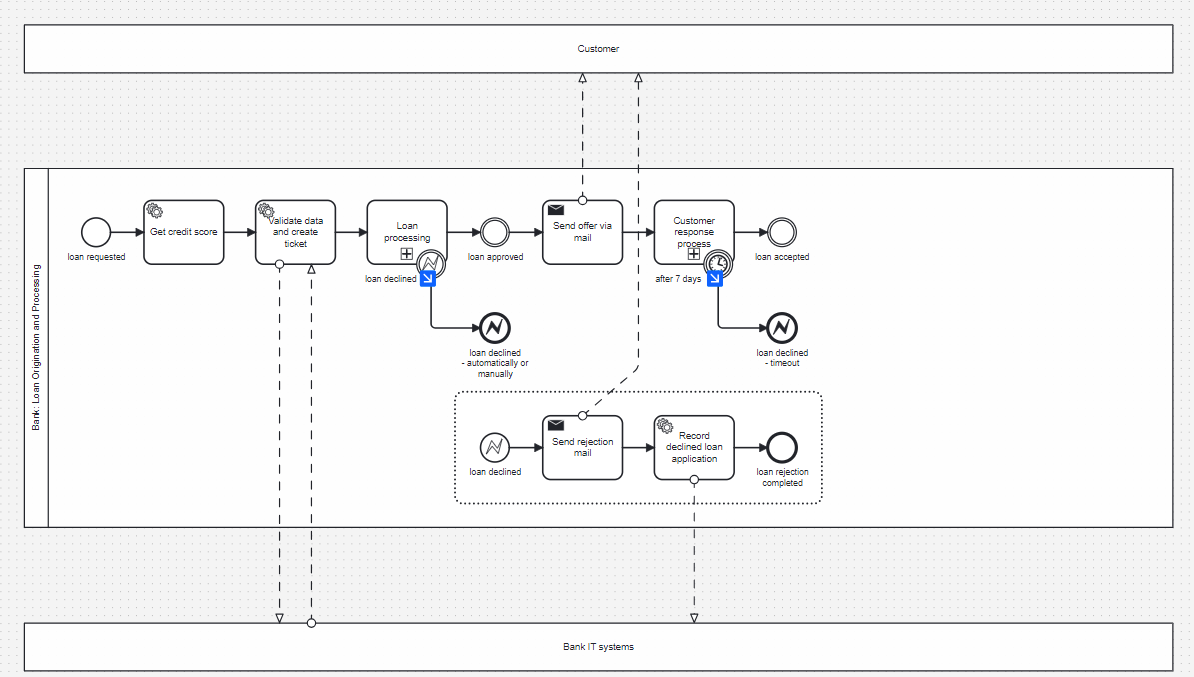
* Номер телефона для связи, отображаемый клиенту в успешном сообщении и письме, должен быть настраиваемым.
* Номер хранится в базе данных и может быть изменен через админ-панель.

### 7. ****Логирование и обработка ошибок****

#### Требования:

* Все действия должны логироваться (валидные данные, решения по займам, ошибки).
* В случае ошибки сервер должен возвращать клиенту обоснованное сообщение об ошибке, чтобы фронтенд мог отобразить его пользователю.

**Диаграмма процесса в нотации BPMN (выполнено в Camunda)**



**Дополнительное задание: в свободной форме предложить идеи по улучшению процесса**

**-** добавитьчат бот для улучшения взаимодействия клиента с системой (напр. чат бот заполнит все необходимые поля на базе информации, предоставленной со стороны клиента) (nice to have)

- добавить смс аутентификацию по номеру клиента, чат бот проверяет действительно ли это клиент предоставил данные (must have)

В задании MVP система для выдачи годовых займов потенциальным клиентам нет описания каким образом клиент будет подписывать документы и получать деньги после одобрения займа: удаленно либо в отделении кредитной организации.

Если планируется выдавать займ удаленно, то необходимо включить следующие операции:

- добавить галочку – клиент согласен на просмотр кредитной истории (must have)

- добавить аутентификацию клиента – клиент делает снимок лица и предоставляет сканы удостоверения личности, чат бот проверяет действительно ли это клиент (must have)

- добавить получение согласия на просмотр кредитной истории (смс аутентификация) (must have)

FICO agencies ниже в диаграмме это ПКБ (Первое кредитное бюро - где хранится кредитная история клиентов)

(два последних действия следует добавить после заполнения первой страницы и нажатия кнопки отправить)

- операция Provide loan (см ниже) можно расширить дополнительной операцией – отправка денег на карточку клиента. Для этого от клиента потребуется дополнительные данные – реквизиты карточки. В целом надо оценить cost value такого решения, расходы по отправке денег на карту клиента можно включить в сумму займа клиента, в таком случае клиента необходимо об этом уведомить заранее, плюс прописать таковые расходы в кредитном контракте. Данная опция также требует дополнительного рисерча процесса выпуска кредитных денег клиенту, поскольку на момент заполнения документа неизвестно как деньги отправляются клиенту (открывается карточка, на которую отправляются деньги, либо деньги выдаются кэшом в кассе кредитной организации, либо деньги отправляются на карточку клиента)

- для удаленной выдачи займа и отправки кредитных денег потребуется от клиента подписать удаленно кредитные документы, для осуществления этой операции нужен рисерч каким образом клиент будет подписывать эти документы – ЭЦП (выпущенная Egov), свое ЭЦП (на примере Каспи, Фридом Финанс либо Altyn Bank) (must have)

