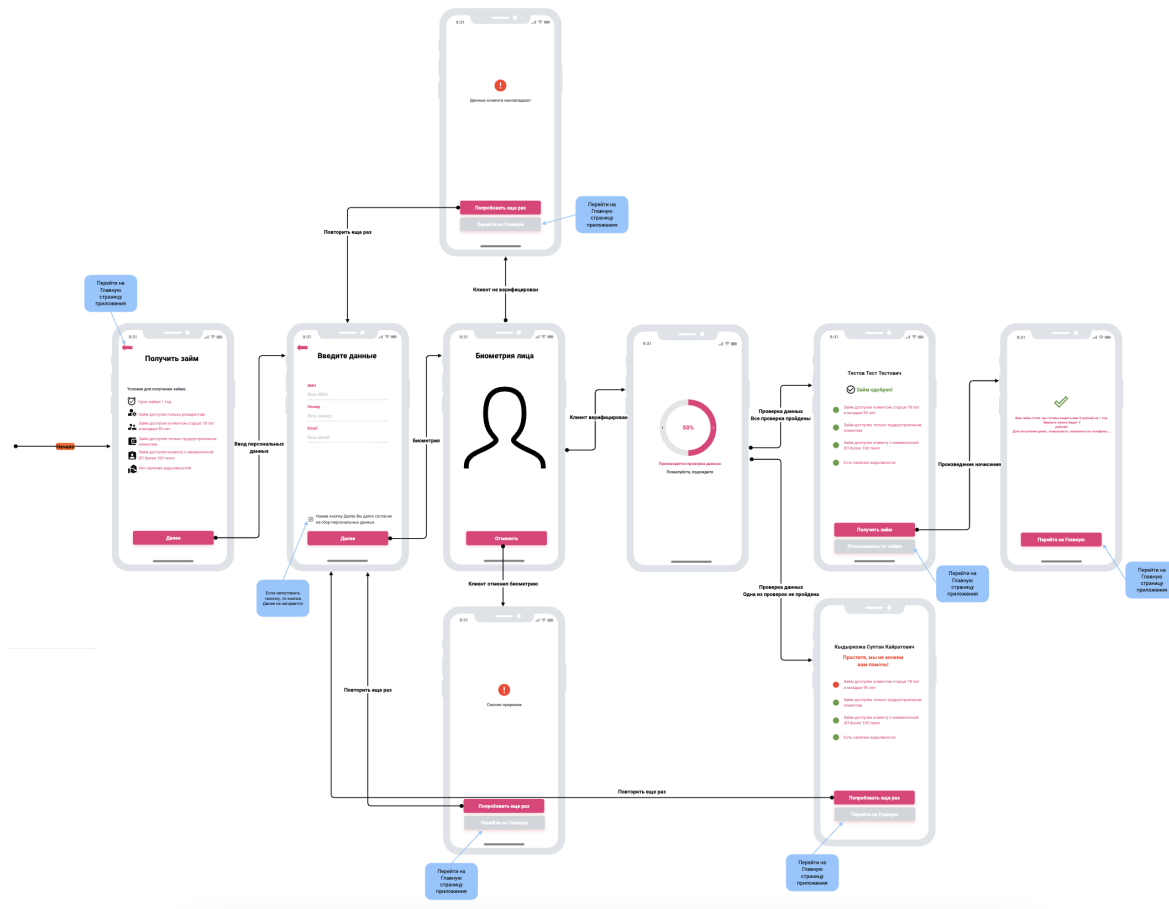


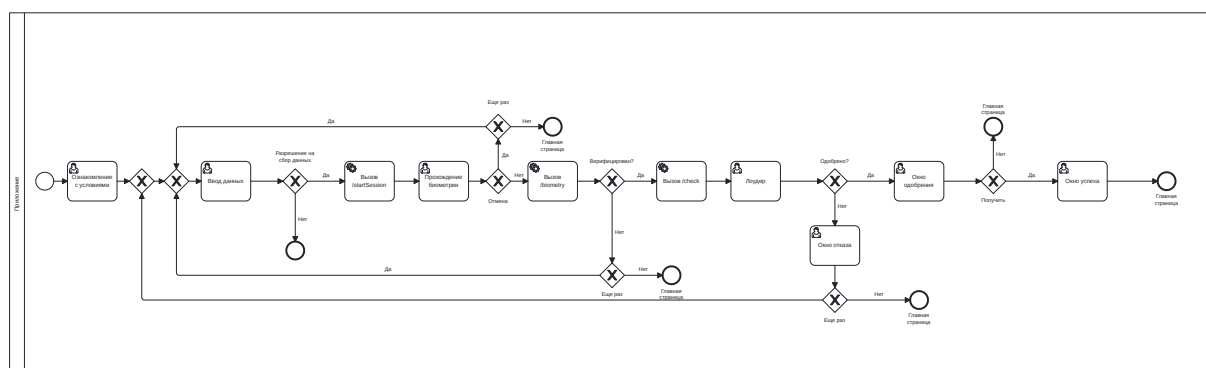
Тестовое задание

1. Цель: создать MVP системы для оценки и выдачи годовых займов клиентам.
2. Требования: *test_assignment.pdf*
3. Мокап:

<https://app.moqups.com/a9BxZkaBZWz61n3paYPLV56Uhcmk4Af1/view/page/a7bc758b4>



4. BPMN:



5. Необходимо предварительно реализовать T3 Backend части

T3 выдачи годовых займов(Backend)

Описание метода /getSession

1. Необходимо реализовать метод /getSession для формирования идентификатора сессии.

а. Описание метода:

i. Входные параметры:

Параметр	Тип	Описание
IIN	string	ИИН
phoneNumber	string	Номер телефона
email	string	Email

ii. Выходные параметры:

Параметр	Тип	Описание
sessionId	string	Идентификатор сессии

iii. Ошибки:

1. Если метод вернул 500-ю ошибку, необходимо формировать код ошибки "SERVICE_ERROR".
2. Если метод вернул 400-ю ошибку, необходимо формировать код ошибки "BAD_REQUEST".
3. Если метод вернул 200, необходимо записать в таблице **client_loan_info** данные из запроса по созданному sessionId:

а. Таблица содержит следующие параметры:

- i. %id% = идентификатор записи в таблице
- ii. %sessionId% = идентификатор сессии
- iii. %IIN% = ИИН
- iv. %phoneNumber% = номер телефона
- v. %email% = Email

Описание метода /biometry

1. Необходимо реализовать метод /biometry для сверки живости и эталонного фото из государственных баз.

а. Описание метода:

i. Входные параметры:

Параметр	Тип	Описание
IIN	string	ИИН
meta_data	string	Мета-данные пройденной биометрии
sessionId	string	Идентификатор сессии

ii. Выходные параметры:

Параметр	Тип	Описание
verify	boolean	Идентифицирован ли клиент TRUE (Да) FALSE (НЕТ)

iii. Ошибки:

1. Если метод вернул 500-ю ошибку, необходимо формировать код ошибки "SERVICE_ERROR".
2. Если метод вернул 400-ю ошибку, необходимо формировать код ошибки "BAD_REQUEST".

б. Логика работы метода:

- i. Для обращения в ГОС базу (А) необходимо сделать интеграцию систем А и В (бэк приложения). Запросы в сторону ГОС баз будут производиться по REST API.

1. По полученному ИИН необходимо сделать запрос в ГОС базу для получения эталонного фото клиента.
2. После получения эталонного фото необходимо произвести сверку фото живости из параметра meta_data с эталонным фото:
 - а. Если процент схожести меньше среднего, то параметр verify принимает значение FALSE.
 - б. Если процент схожести больше среднего, то параметр verify принимает значение TRUE.

- ii. При получении запроса необходимо по параметру sessionId сохранить данные в таблице **client_loan_info**.

1. Таблица содержит следующие параметры:
 - а. %meta_data% = Мета-данные пройденной биометрии.

- iii. После получения результата схожести необходимо по sessionId сохранить данные в таблице **client_loan_info**:

1. Таблица содержит следующие параметры:

- a. %verify% = Идентифицирован ли клиент.

Описание метода /checkClientEligibility

1. Вызывается метод /checkClientEligibility для проверки доступности займа для клиента.

a. Описание метода:

i. Входные параметры:

Параметр	Тип	Описание
IIN	string	ИИН
sessionId	string	Идентификатор сессии

ii. Выходные параметры:

Параметр	Тип	Описание
AGE_ELIGIBLE	boolean	Клиент подходит по возрасту
WAGES_ELIGIBLE	boolean	Клиент подходит по зарплате
WORK_ELIGIBLE	boolean	Клиент трудоустроен
DEBT_ELIGIBLE	boolean	У клиента нет задолженностей

iii. Ошибки:

1. Если метод вернул 500-ю ошибку, необходимо формировать код ошибки "SERVICE_ERROR".
2. Если метод вернул 400-ю ошибку, необходимо формировать код ошибки "BAD_REQUEST".

b. Логика работы метода:

i. При наличии интеграции с ГОС базой необходимо по ИИН получить следующие данные:

1. ФИО.

2. Дата рождения:

- a. На основе данного параметра необходимо вычислить возраст клиента по формуле: текущий год - год рождения = возраст.

i. Если возраст более 18 и менее 90 лет, то параметр AGE_ELIGIBLE принимает значение TRUE.

- ii. Если возраст менее 18 или более 90 лет, то параметр AGE_ELIGIBLE принимает значение FALSE.
- 3. Наличие официального трудоустройства:
 - a. Если клиент официально трудоустроен, параметр WORK_ELIGIBLE принимает значение TRUE.
 - b. Если клиент официально не трудоустроен, параметр WORK_ELIGIBLE принимает значение FALSE.
- 4. Ежемесячная заработная плата:
 - a. Если зарплата больше 100 тенге, параметр WAGES_ELIGIBLE принимает значение TRUE.
 - b. Если зарплата меньше 100 тенге или параметр WORK_ELIGIBLE = FALSE, параметр WAGES_ELIGIBLE принимает значение FALSE.
- 5. Место работы.
- 6. Отсутствие задолженностей:
 - a. Если задолженности нет, параметр DEBT_ELIGIBLE принимает значение TRUE.
 - b. Если задолженность есть, параметр DEBT_ELIGIBLE принимает значение FALSE.
- ii. После получения данных необходимо сохранить их в таблице **client_loan_info** по параметру sessionId:
 - 1. Таблица содержит следующие параметры:
 - a. %name% = ФИО.
 - b. %birth_date% = Дата рождения.
 - c. %work_on% = Наличие официального трудоустройства.
 - d. %wages% = Заработная плата.
 - e. %work_place% = Место работы.
 - f. %not_debt% = Нет задолженностей.
- iii. Расчет кредитного предложения:
 - 1. Если клиенту не был дан отказ (все параметры ELIGIBLE равны TRUE), рассчитывается сумма займа по формуле:
 Сумма займа = (Зарплата –
 Месячные выплаты по другим задолженностям) × 10
 Сумма займа = (Зарплата – Месячные выплаты по другим задолженностям) × 10

2. Рассчитывается сумма для возврата с учетом процентной ставки, которая составляет 3% в месяц от суммы основного долга:

Сумма возврата = Сумма займа + (Сумма займа × 0.03)

Сумма возврата=Сумма займа+(Сумма займа×0.03)

3. Допустим, у нас есть следующие данные по клиенту:

- Зарплата: 300 000 тенге
- Месячные выплаты по другим задолженностям: 50 000 тенге
- Все параметры проверки (AGE_ELIGIBLE, WAGES_ELIGIBLE, WORK_ELIGIBLE, DEBT_ELIGIBLE) вернули TRUE, что значит, что отказ не был дан.

Расчет суммы займа:

Сумма займа = (Зарплата –

Месячные выплаты по другим задолженностям) × 10 = (300000 – 50000) × 10 = 250000 × 10 = 2500000 тенге

Расчет суммы для возврата с учетом процентной ставки 3%:

Сумма возврата = Сумма займа + (Сумма займа × 0.03) = 2500000 + (2500000 × 0.03) = 2500000 + 75000 = 2575000 тенге

Таким образом:

- Сумма займа: 2 500 000 тенге
- Сумма для возврата: 2 575 000 тенге

- iv. После формирования ответа необходимо результат сохранить в формате JSON в таблице **client_loan_info** в поле %result%.

Описание метода /sendEmail

1. Вызывается метод /sendEmail для отправки решения на указанный email:

- a. Описание метода:

Параметр	Тип	Описание
send_from	string	Отправитель
send_to	string	Получатель
text	string	Текст решения
sessionId	string	Идентификатор сессии

- b. Выходные параметры:

- i. HTTP 200 Success.

- ii. HTTP 400 Bad Request.
- iii. HTTP 500 Server Error.
- с. Ошибки:
 - i. Если метод вернул 500-ю ошибку, необходимо формировать код ошибки "SERVICE_ERROR".
 - ii. Если метод вернул 400-ю ошибку, необходимо формировать код ошибки "BAD_REQUEST".
- 2. После отправки данных необходимо записать в таблице **client_loan_info** данные об отправке решения по почте:
%send% = TRUE/FALSE
- 6. После на основе реализации Backend реализовать T3 Frontend части

T3 выдачи годовых займов(Frontend)

Use-case: Клиент зашел в раздел получения займа через мобильное приложение

Клиент заходит в приложение → Главная страница → раздел "Получить годовой займ".

1. Клиент ознакомливается с "Условиями получения займа":
 - a. Срок займа — 1 год.
 - b. Займ доступен только резидентам.
 - c. Займ доступен клиентам старше 18 лет и младше 90 лет.
 - d. Займ доступен только трудоустроенным клиентам.
 - e. Займ доступен клиентам с ежемесячной зарплатой более 100 тенге.
 - f. У клиента не должно быть задолженностей.
2. Клиент нажимает кнопку "Далее", происходит переход на страницу ввода данных клиента:
 - a. Если клиент не ввел все необходимые данные, кнопка "Далее" - не активна
 - b. Клиент вводит данные:
 - i. **ИИН — проверки будут проводиться по введенному ИИН:**
 1. Поле обязательное.
 2. Необходимо сделать проверку на количество символов (12).
 - ii. **Номер — на указанный номер будут приходить СМС-уведомления по займу:**

1. Поле обязательное.
 2. Необходимо сделать проверку на количество символов (10).
 3. Необходимо вшить +7 на фронте.
- iii. **Email — на указанный email будет отправлено решение по займу:**
1. Поле обязательное.
- iv. Галочка для согласия обработки данных
1. Если клиент не нажал на галочку, кнопка "Далее" - не активна
- с. Клиент нажимает кнопку "Далее":
- i. Вызывается метод /startSession для формирования идентификатора сессии:
 1. Описание метода:
 - a. Входные параметры:

Параметр	Тип	Описание
IIN	string	ИИН
phoneNumber	string	Номер
email	string	Email
 - b. Выходные параметры:

Параметр	Тип	Описание
sessionId	string	Идентификатор сессии
 - c. Если метод вернул 500-ю ошибку, необходимо по коду ошибки отобразить клиенту бизнесовую ошибку.
 - d. Если метод отработал успешно, необходимо перейти на следующий шаг.
3. Открывается страница прохождения биометрии (при наличии внутреннего движка биометрии), клиент проходит биометрию:
- a. Если клиент не прошел верификацию, отображается страница "Данные клиента не совпадают" с возможностью попробовать еще раз или перейти на главную страницу:
 - i. Если клиент хочет попробовать еще раз, нажимает на кнопку "Попробовать еще раз", и его переводят на страницу ввода данных.
 - ii. Если клиент хочет завершить сессию, он нажимает на кнопку "Перейти на Главную".
 - b. Если клиент при прохождении биометрии нажал на кнопку "Отменить", отображается страница "Сессия прервана" с возможностью попробовать

еще раз или перейти на главную страницу:

- i. Если клиент хочет попробовать еще раз, он нажимает на кнопку “Попробовать еще раз”, и его переводят на страницу ввода данных.
 - ii. Если клиент хочет завершить сессию, он нажимает на кнопку “Перейти на Главную”.
- c. Вызывается метод /biometry для сверки живости и эталонного фото из государственных баз:
- i. Описание метода:

1. Входные параметры:

Параметр	Тип	Описание
IIN	string	ИИН
meta_data	string	Мета-данные пройденной биометрии
sessionId	string	Идентификатор сессии

2. Выходные параметры:

Параметр	Тип	Описание
verify	boolean	Идентифицирован ли клиент TRUE (Да) FALSE (Нет)

3. Если метод вернул 500-ю ошибку, необходимо по коду ошибки отобразить клиенту бизнесовую ошибку.
 4. Если verify = TRUE, необходимо перейти на следующий шаг.
 5. Если verify = FALSE, необходимо по коду ошибки отобразить клиенту бизнесовую ошибку.
- d. Вызывается метод /checkClientEligibility для проверки доступности займа для клиента

i. Описание метода:

1. Входные параметры:

Параметр	Тип	Описание
IIN	string	ИИН
sessionId	string	Идентификатор сессии

2. Выходные параметры:

Параметр	Тип	Описание
AGE_ELIGIBLE	boolean	Клиент подходит по возрасту

WAGES_ELIGIBLE	boolean	Клиент подходит по зарплате
WORK_ELIGIBLE	boolean	Клиент трудоустроен
DEB_ELIGIBLE	boolean	У клиента нет задолженностей

3. Если метод вернул 500-ю ошибку, необходимо по коду ошибки отобразить клиенту бизнесовую ошибку.
 4. Если метод отработал успешно, необходимо перейти на следующий шаг.
4. Клиенту отображается загрузчик, после получения ответа фронт отображает результат:
- a. Если клиент проходит по всем вышеуказанным параметрам, он может получить займ или отказаться:
 - i. Если клиент отказывается от займа, он нажимает на кнопку "Отказываюсь от займа", и его переводят на главную страницу.
 - ii. Если клиент принимает займ, он нажимает на кнопку "Получить займ", и его переводят на экран успеха.
 - b. Если клиент не проходит по всем вышеуказанным параметрам, он может попробовать еще раз или перейти на главную страницу:
 - i. Если клиент хочет попробовать еще раз, он нажимает на кнопку "Попробовать еще раз", и его переводят на страницу ввода данных.
 - ii. Если клиент хочет завершить сессию, он нажимает на кнопку "Перейти на Главную".
5. Клиенту отображается страница успеха с возможностью перехода на главную страницу:
- a. Вызывается метод /sendEmail для отправки решения на указанный email:
 - i. Описание метода:

Параметр	Тип	Описание
send_from	string	Отправитель
send_to	string	Получатель
text	string	Текст решения

- ii. Выходные параметры:
 1. HTTP 200 Success.
 2. HTTP 400 Bad Request.
 3. HTTP 500 Server Error.

Заключение

Все данные по пользователю, такие как возраст, трудоустройство, заработная плата и наличие задолженности, были получены через ИИН. Локализация не была учтена — необходимо, чтобы фронтенд часть получала параметр, определяющий язык отображения информации. Регистрация номера была выполнена без использования ОТР.

Идеи по улучшению процесса:

1. **Предварительная оценка займа:** Добавить возможность быстрой предварительной оценки займа на основе введенных данных до полной подачи заявки. Это сократит время ожидания для клиентов, которым может быть отказано.
2. **Автоматизация проверки данных:** Интегрировать систему с государственными или кредитными базами данных для автоматической проверки трудоустройства и задолженностей клиента, чтобы исключить человеческие ошибки и ускорить процесс.
3. **Гибкость условий займа:** В зависимости от профиля клиента предлагать не только фиксированные условия займа, но и различные варианты (например, с разными сроками или ставками).
4. **Персонализация предложений:** На основе анализа данных клиентов можно предлагать персонализированные условия займа, например, с уменьшенной ставкой для надежных клиентов или увеличением суммы займа для клиентов с высоким доходом.
5. **Оценка кредитного рейтинга:** Добавить интеграцию с кредитными бюро для получения кредитного рейтинга и его использования при расчете возможного займа.
6. **Аналитика и улучшение модели:** Постепенно улучшать модель оценки займов на основе анализа данных о поведении клиентов и результатах выданных займов, чтобы точнее предсказывать риски и предоставлять более выгодные условия для надежных клиентов.