

Документация к продукту SPB-BANK (Svetlana).

	SCRUM-79
Ссылка на проект	SCRUM-4
Статус страницы	<u>В работе</u>
Тестируемый сайт	https://demo.bspb.ru/

▼ Тестируемые фичи:

1. Поля ввода логина и пароля (начальная страница);
2. Поле ввода код из смс;
3. Отправка реферальной ссылки;
4. Экран страхование автомобиля;
5. Экран актуальные документы.

Описание требований

1. Требования к полю логин/пароль:

1.1. Логин: длинна 2-20 символов, нижний регистр, латиница.

1.2. Пароль: 2-10 символов, нижний регистр, латиница.

1.3. При вводе валидных логина/пароля и нажатии кнопки “Войти”, осуществляется переход

на страницу ввода “Код из смс”.

1.4. При введении невалидных данных даётся 3 попытки авторизации.

2. Требования к полю код из смс:

2.1. Номер для смс должен быть верифицирован в настройках аккаунта.

2.2. Код подтверждения смс - 4 символа, содержит только цифры.

2.3. Повторная отправка кода возможна через 120 сек., даётся 3 попытки.

2.4. При вводе валидного кода из смс и нажатии кнопки “Войти” , осуществляется переход на главную страницу личного кабинета.

3. Требования к заполнению поля email при отправке реферальной ссылки:

3.1. При отправке пустого поля email, всплывает ошибка: обязательно для заполнения.

3.2. Длина 2-20 символов, наличие домена, нижний регистр, спец.символы - @, точка.

3.3. При вводе неверных данных поле подсвечивается красным и всплывает ошибка

“Неверный ввод данных”.

4. Экран страхование автомобилей.

4.1. При нажатии должна появиться выпадающая шторка с разделами: “Страхование автомобиля”, “Страхование жизни, здоровья и спорта”, “Защита от задержки или отмены рейса”, “Страхование недвижимости”, “Пенсионное обеспечение”.

4.2. Находясь в любом из разделов (см. выше) при нажатии на кнопку

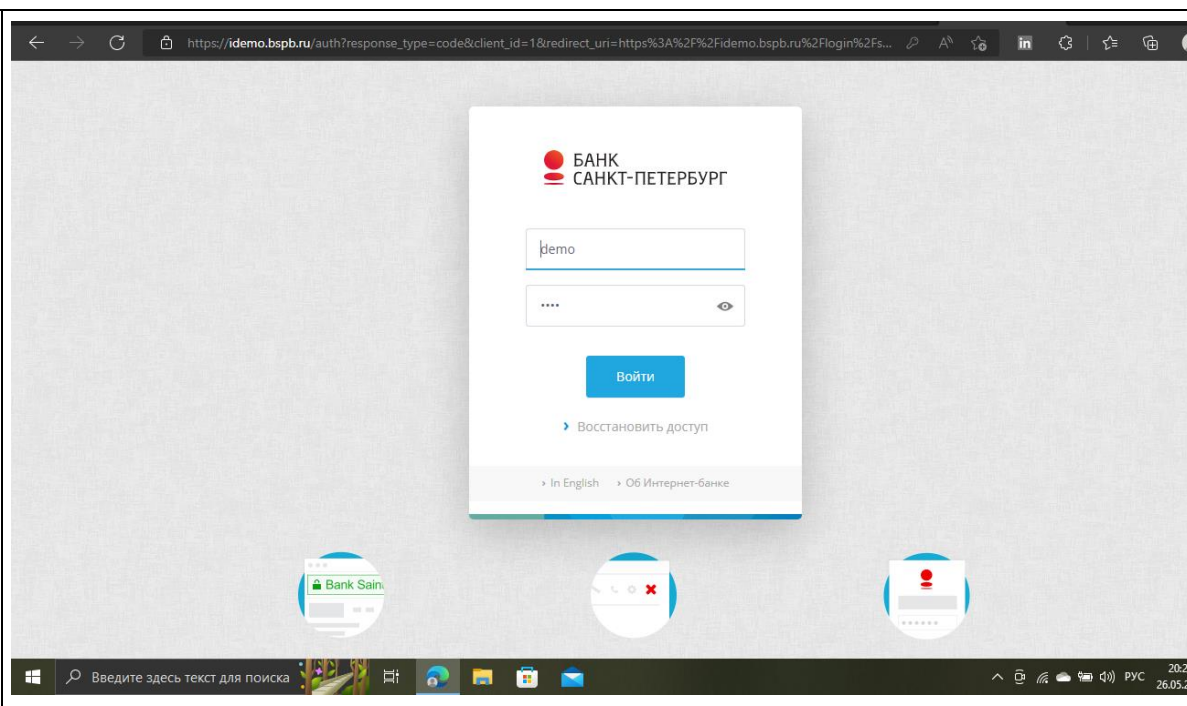
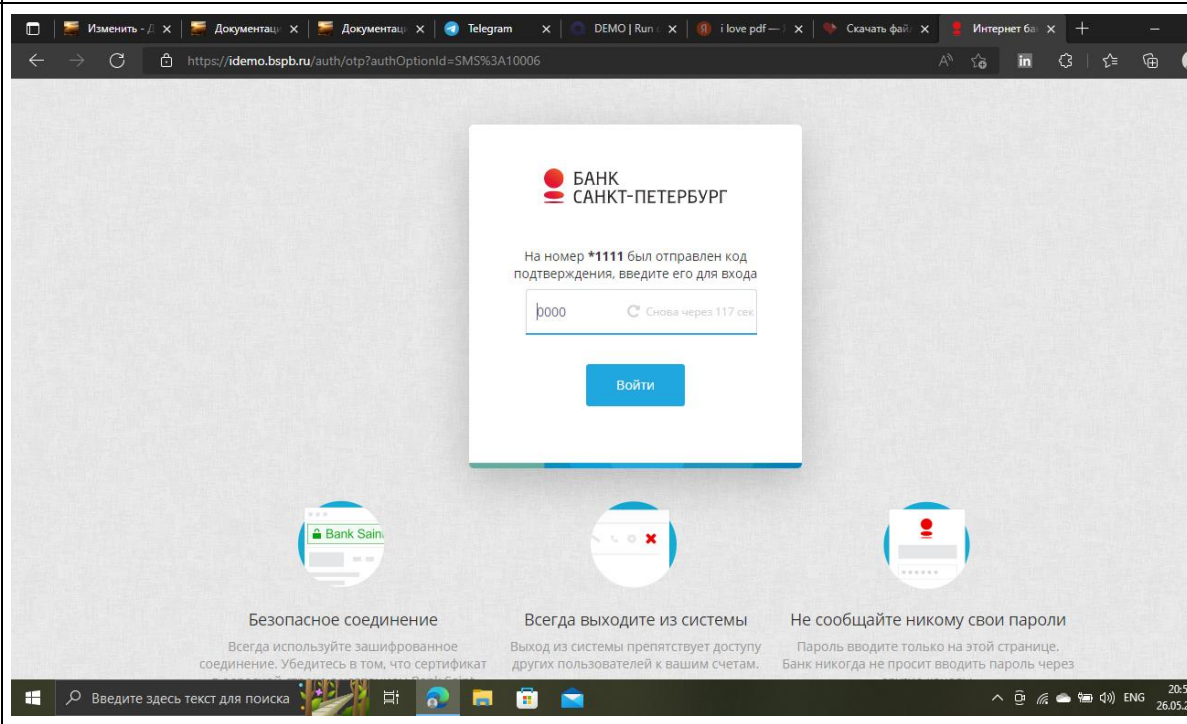
“Оформить страховку”, осуществляется переход на страницу оформления.

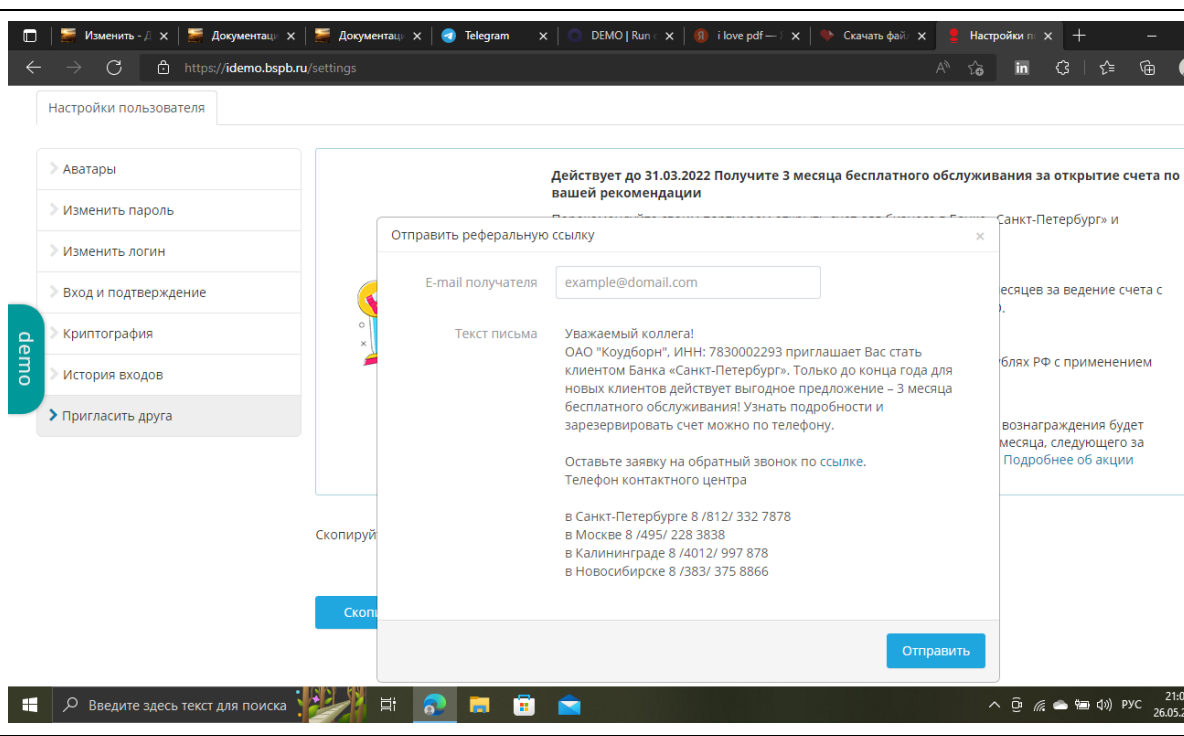
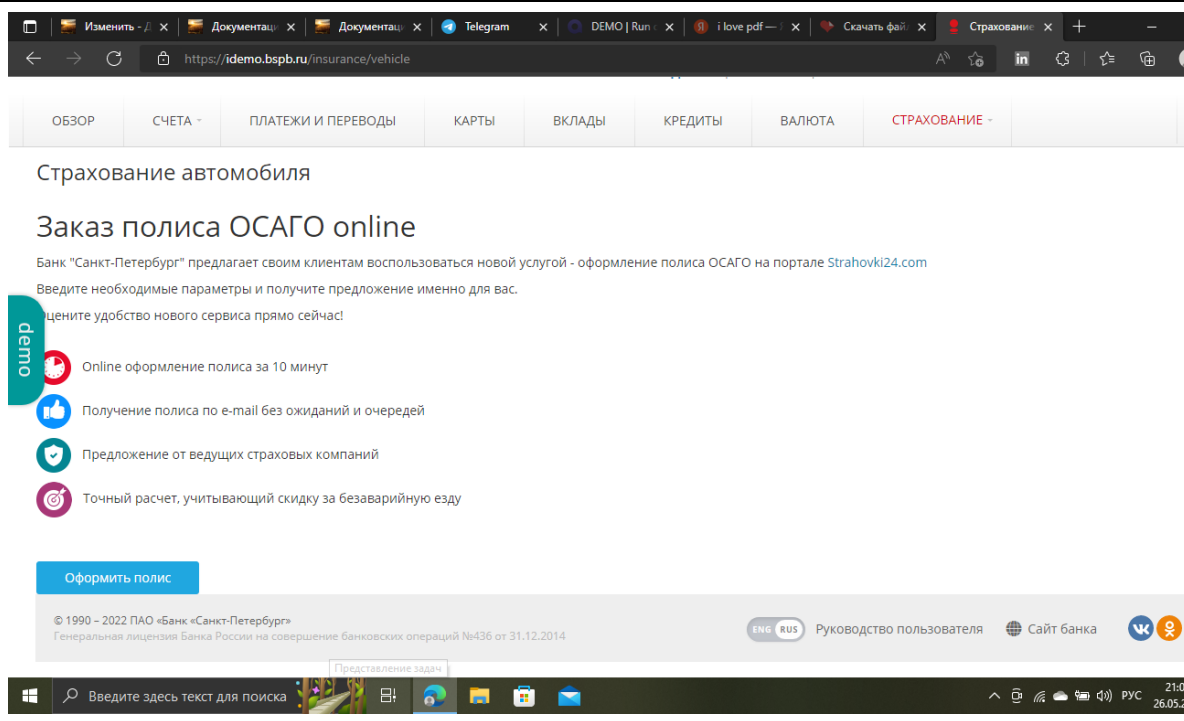
5. Экран актуальные документы.

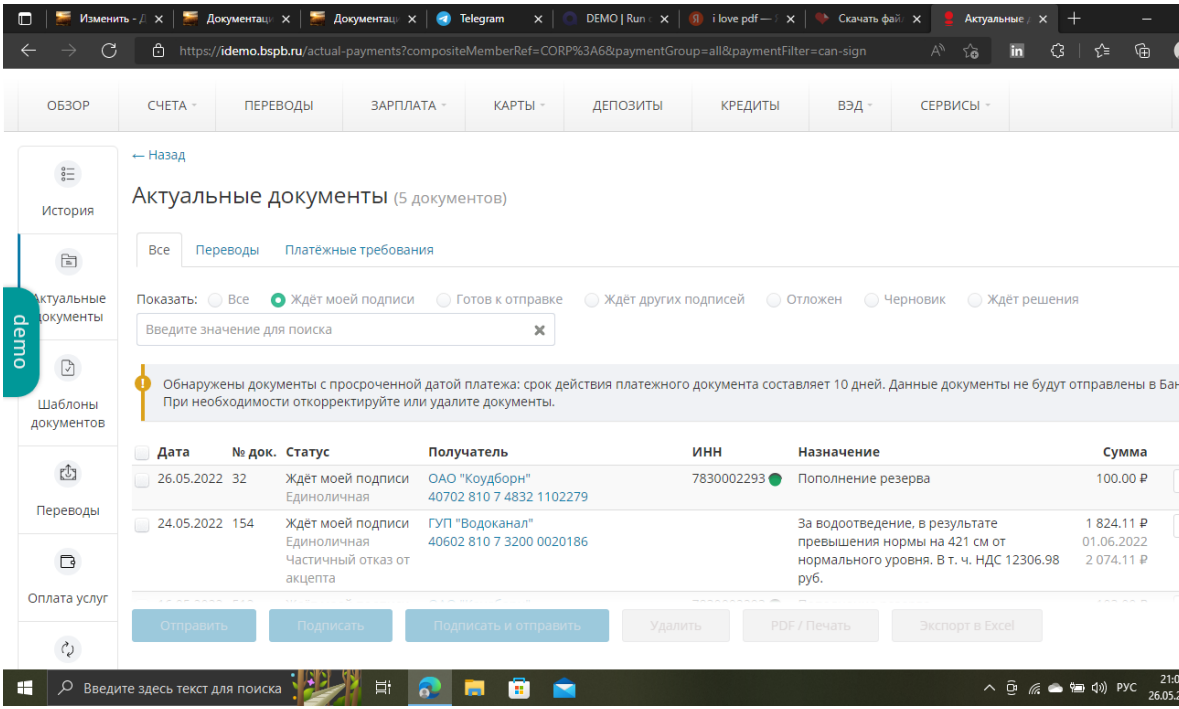
5.1. Внизу слева кнопка “Отправить” должна быть активной.

Макеты к требованиям

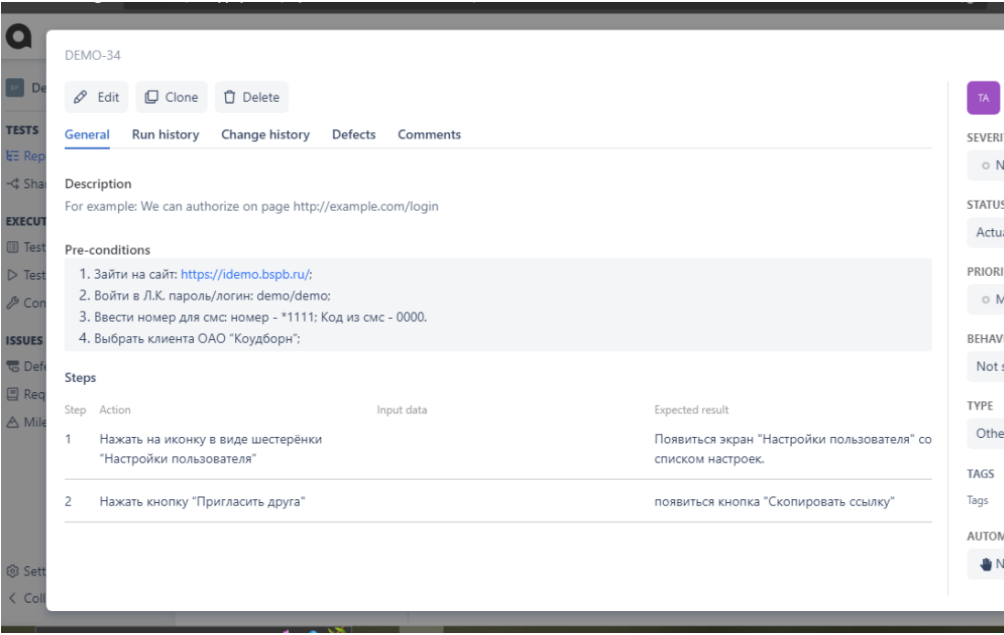
	Название экрана	Макет
--	--------------------	-------

1.	Экран ввода логин/пароль - demo/demo.	
2.	Экран ввода номера для смс: номер - *1111; Код из смс - 0000.	

3.	<p>Реферальная ссылка, поле ввода “Email”.</p>	
4.	<p>Экран настройки пользователя - раздел “Страхование автомобиля”, кнопка “Оформить полис”.</p>	

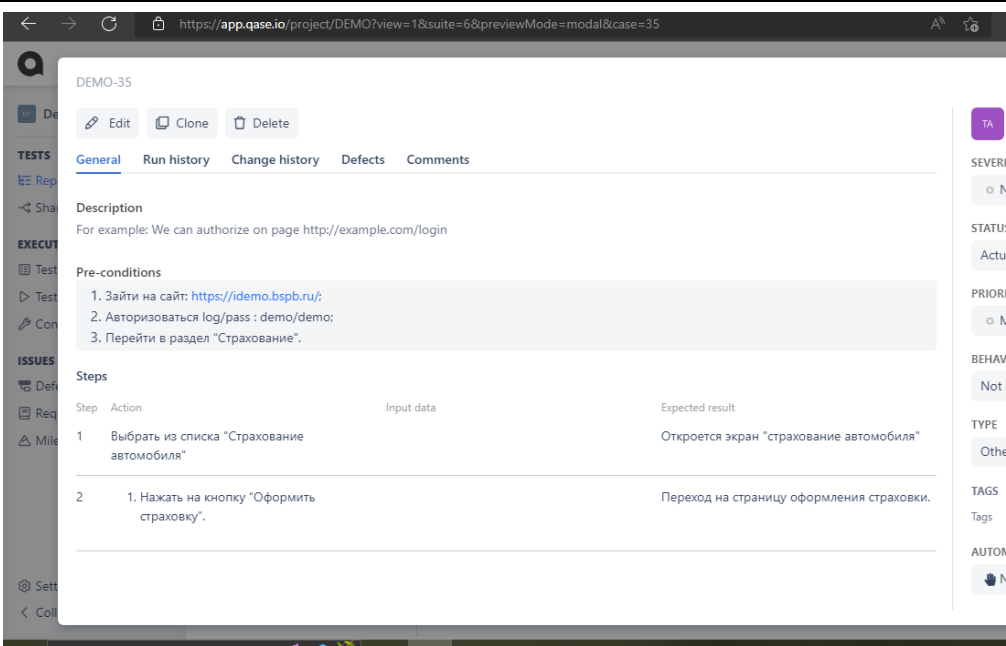
<p>5. Экран актуальные документы.</p>	
---------------------------------------	--

Баг - репорты, тест - кейсы

Описание	Баг - репорт	Тест - кейс
<p>[Экран Настройки пользователя] Не отправляется реферальная ссылка другу.</p>	<p>Доска SCRUM - Agile-доска - Jira (atlassian.net)</p>	

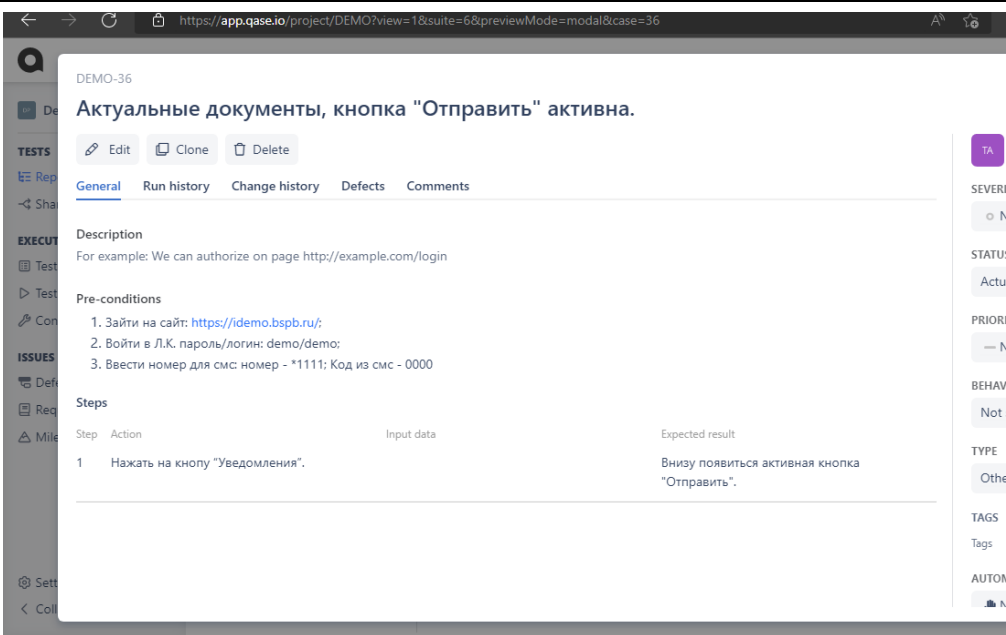
[Экран
страхование
автомобиля]
при нажатии
на кнопку
оформить
страховку
всплывает
роп-уп "Вы
покидаете
сайт..."

[Доска
SCRUM -
Agile-доска -
Jira
\(atlassian.net\)](#)



[Экран
актуальные
документы]
Не активная
кнопка
"Отправить".

[Доска
SCRUM -
Agile-доска -
Jira
\(atlassian.net\)](#)



База данных:

SQL запросы

1. CREATE DATABASE spbank;
2. USE spbank;
3. CREATE TABLE Authorization (
id int NOT NULL primary key,
login VARCHAR(50),

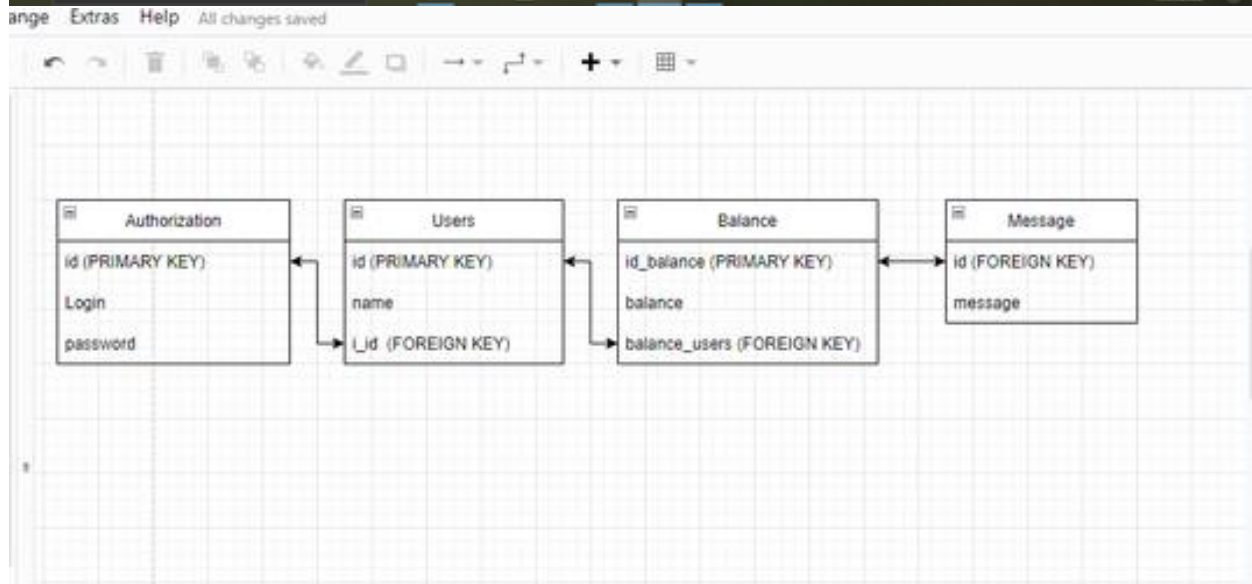
- ```
password VARCHAR(50) NOT NULL
);
```
4. CREATE TABLE Users (  
id INT NOT NULL PRIMARY KEY,  
name VARCHAR(50),  
i\_id INT,  
FOREIGN KEY Users (i\_id) REFERENCES Authorization (id)  
);
  5. CREATE TABLE Balance (  
id\_balance INT NOT NULL PRIMARY KEY,  
balance INT,  
balance\_users INT,  
FOREIGN KEY Balance (balance\_users) REFERENCES Users (id)  
);
  6. CREATE TABLE Message (  
id INT NOT NULL,  
message TEXT,  
FOREIGN KEY Message (id) REFERENCES Users (id)  
);
  7. SHOW TABLES;
  8. INSERT INTO Authorization (id, login, password) VALUES (1, 'Sveta', '1111'), (2, 'Ivan', '2222'), (3, 'Mikl', '3333'), (4, 'Valentina', '4444'), (5, 'Irina', '5555'), (6, 'Petr', '6666'), (7, 'Alex', '7777');
  9. INSERT INTO Users (id, name, i\_id) VALUES (1, 'Setlana\_Vlushina', 1), (2, 'Ivan\_Ivanov', 2), (3, 'Mikl\_Demidov', 3), (4, 'Valentina\_Romanova', 4), (5, 'Irina\_Demidova', 5), (6, 'Petr\_Yasnetsov', 6), (7, 'Alex\_Alex', 7);
  10. INSERT INTO Balance (id\_balance, balance, balance\_users) VALUES (1, 50000, 4), (2, 60000, 2), (3, 35000, 3), (4, 75500, 1), (5, 40500, 5), (6, 65000, 6), (7, 40000, 7);
  11. INSERT INTO Message (id, message) VALUES (1, 'Is it possible to cancel the last operation on the card'), (2, 'How can I change the pin code of the card'), (3, 'Is it possible to reissue the card'), (4, 'Why does not my card balance show'), (5, 'Can I link a child card to this card?');

```
mysql> SELECT * FROM Authorization;
+----+-----+-----+
| id | login | password |
+----+-----+-----+
1	Sveta	1111
2	Ivan	2222
3	Miki	3333
4	Valentina	4444
5	Irina	5555
6	Petr	6666
7	Alex	7777
+----+-----+-----+
7 rows in set (0.00 sec)

mysql> SELECT * FROM Balance;
+-----+-----+-----+
| id_balance | balance | balance_users |
+-----+-----+-----+
1	10000	4
2	20000	2
3	30000	3
4	75000	1
5	40000	5
6	65000	6
7	80000	7
+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> SELECT * FROM Message;
+----+-----+
| id | message |
+----+-----+
1	Is it possible to cancel the last operation on the card
2	How can I change the pin code of the card
3	Is it possible to reissue the card
4	Why does not my card balance show
5	Can I link a child card to this card?
+----+-----+
5 rows in set (0.00 sec)

mysql>
```



## Bash

### Команды

1. Перейти в домашнюю директорию - `cd /;`
2. Создать новую папку - `mkdir TestProject;`
3. Перейти в созданную папку - `cd TestProject ;`
4. Создать в этой папке новый файл - `touch newProject.txt;`
5. Перейти в текстовый редактор - `vim newProject.txt ;`
6. Выйти и сохранить этот текст - `Esc :wg;`



7. Посмотреть конкретное словосочетание в файле - `cat newProject.txt | grep отлично`;
8. Посмотреть конкретное словосочетание в файле и перенаправить поток в новый файл - `cat newProject.txt | grep отлично >> "новый файл"`;
9. Посмотреть, что находится в вашей директории - `ls`;
10. Посмотреть, что находится в вашей директории, но, включая скрытые файлы - `ls-a`;
11. Посмотреть, где вы находитесь (в плане пути) - `pwd`;
12. Вы в домашней директории > создать 2 папки > перейти в первую любую папку > создать там файл > скопировать и перенести этот файл во вторую папку -  
`mkdir Test1, Test2 > cd Test1`

## Postman

### ✓ Коллекция

#### → PetPost

POST <https://petstore.swagger.io/v2/pet> [200 OK, 594B, 645ms]

1. Response time is less than 200ms  
✓ Successful POST request
2. Status code name has strin

#### → PetGet

GET {{pet}}pet/779 [errored]

getaddrinfo ENOTFOUND {{pet}}pet

4. Body is correct

5. Response time is less than 200ms

#### → PutPet

PUT {{pet}}pet [errored]

getaddrinfo ENOTFOUND {{pet}}pet

7. Check response

8. Status code is 200

9. Response time is less than 200ms

#### → DeletePet

DELETE {{pet}}pet/779 [errored]

getaddrinfo ENOTFOUND {{pet}}pet

11. Successful DELETE response

12. Response time is less than 200ms

13. Body matches string

| total run duration: 1193ms

| total data received: 268B (approx)

| average response time: 645ms [min: 645ms, max: 645ms, s.d.: 0µs]

|                    | executed | failed |
|--------------------|----------|--------|
| iterations         | 1        | 0      |
| requests           | 4        | 3      |
| test-scripts       | 4        | 0      |
| prerequest-scripts | 0        | 0      |
| assertions         | 11       | 10     |

Запуск с командной строки

Будь паря C:\WINDOWS\system32\cmd.exe

```
^C
C:\Users\acer>newman run https://www.getpostman.com/collections/c724f2833358752d59a5
newman

PetApi3004

→ CreatePet
POST https://petstore.swagger.io/v2/pet [200 OK, 597B, 611ms]
✓ Status code is 200
✓ Successful POST request
✓ Response time is less than 200ms
✓ Status code name has string

→ IdPet
GET https://petstore.swagger.io/v2/pet/826 [200 OK, 597B, 144ms]
✓ Status code is 200
✓ Response time is less than 200ms
✓ Status code name has string
✓ Content-Type is present

→ PutPet
PUT https://petstore.swagger.io/v2/pet [200 OK, 603B, 148ms]
✓ Status code is 200
✓ Response time is less than 200ms
✓ Status code name has string
✓ Content-Type is present

→ DelPet
DELETE https://petstore.swagger.io/v2/pet/826 [200 OK, 371B, 134ms]
✓ Status code is 200
✓ Response time is less than 200ms
✓ Status code name has string
✓ Body is correct
```

|                    | executed | failed |
|--------------------|----------|--------|
| iterations         | 1        | 0      |
| requests           | 4        | 0      |
| test-scripts       | 4        | 0      |
| prerequest-scripts | 0        | 0      |
| assertions         | 16       | 0      |

Json

```
▼ json
{
 "firstName": "Sveta",
 "lastname": "Svetlay",
 "age": "33",
 "address": {
 "streeetAddress": "Sretenka",
 "city": "Moscow",
 "country": "Russian Federation"
 },
 "phoneNumbers": [
 {
 "type": "home",
 "number": "49500000000"
 }
]
}
```

```

 },
 {
 "type": "mobile",
 "number": "89998880000"
 }
]
}

```

The screenshot shows the JSON Path Finder interface. On the left, a JSON object is displayed with line numbers 1 through 21. The object contains personal information and a list of phone numbers. On the right, the 'Path: x.phoneNumbers[1]' is entered, and the corresponding JSON fragment is shown in a tree view.

**JSON Object:**

```

{
 "firstName": "Sveta",
 "lastName": "Svetlay",
 "age": "33",
 "address": {
 "streetAddress": "Sretenka",
 "city": "Moscow",
 "country": "Russian Federation"
 },
 "phoneNumbers": [
 {
 "type": "home",
 "number": "4950000000"
 },
 {
 "type": "mobile",
 "number": "89998880000"
 }
]
}

```

**Path: x.phoneNumbers[1]**

**Result:**

```

{
 "type": "mobile",
 "number": "89998880000"
}

```

Charles

Скрины

