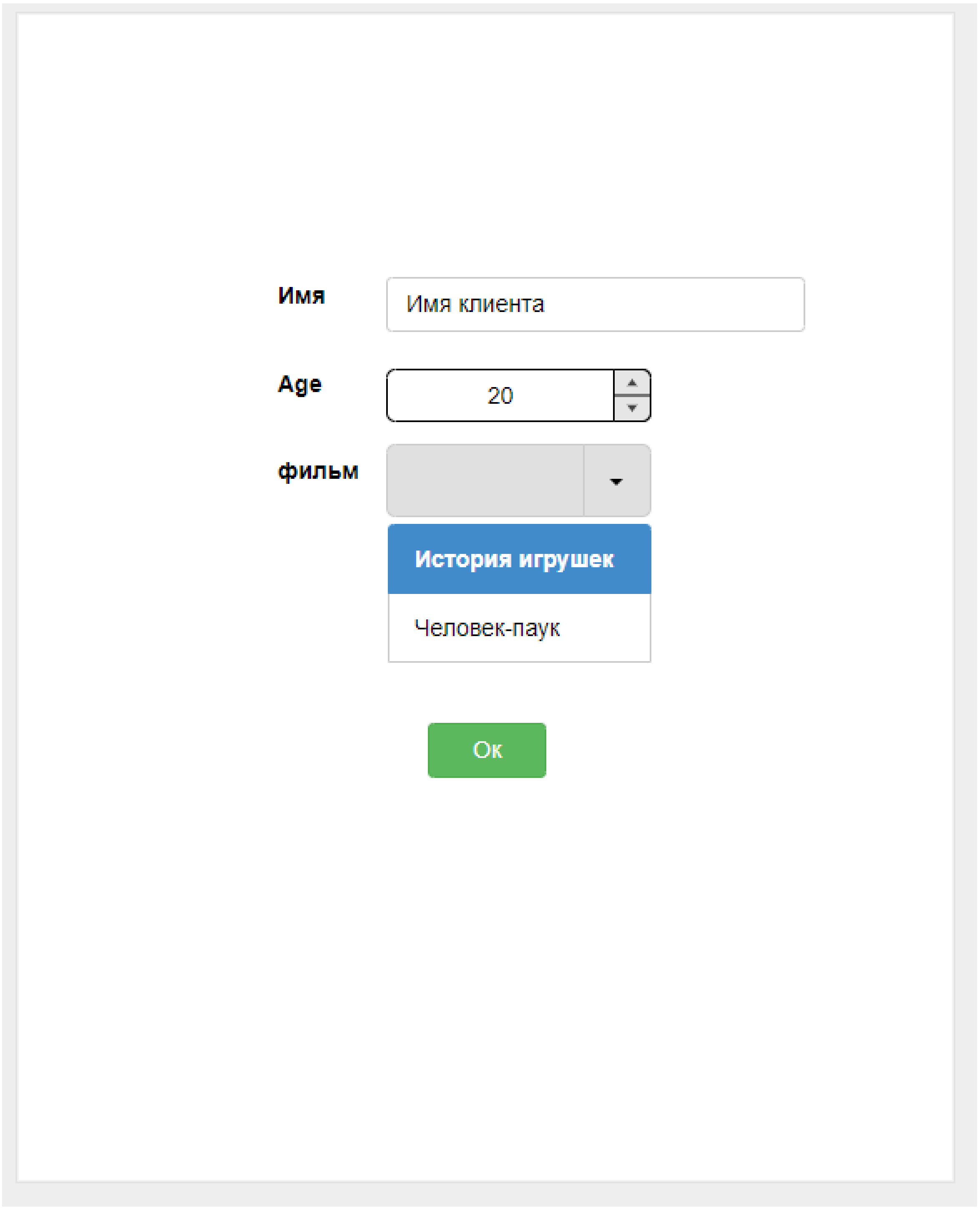
Task №1.



In front of you is a web form of a browser-based application for booking movie tickets to a theater. The user enters data and submits the completed form, after which the movie is booked.

You need to make an optimal list of checks that you will perform to test this service, as well as the expected result for each of the checks.

If you find errors on the form - fix them in an arbitrary form.

Parameters (Documentation for the task):

Name - you can enter any name in Russian up to 20 characters.

Age - only numbers can be entered.

Drop-down list "movie" - list of movies:

-Toy Story - no age restrictions.

-Spider-Man - restriction 12+.

Button "Ok" - Passes the data to the server, if the age is specified less than the allowed age for the movie, then when you click the button "Ok" error is displayed "The selected movie "Spider-Man" has an age restriction of 12+".

After completing the task

1.Write what test design and test analysis techniques you used and for what purpose?

2.What tools would you use to test by written checks?

3.What other types of testing would you apply?

4.What information did you lack to test this form in more detail?

Answers:

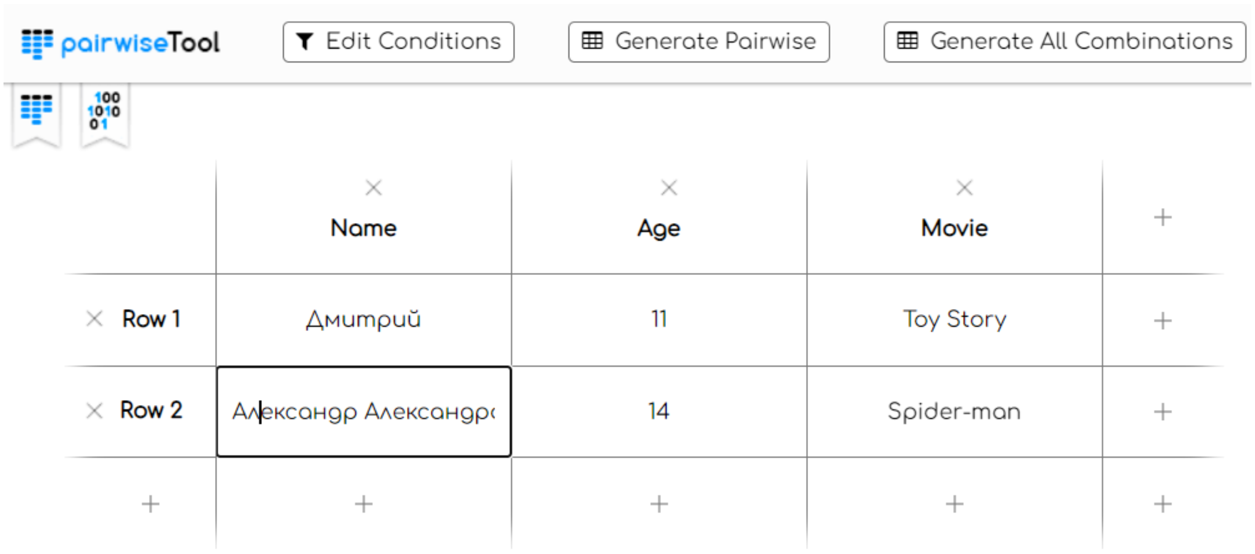
	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8
Age less than 12	+	+	+	+	-	-	-	-
Age from 12	-	-	-	-	+	+	+	+
Entering the correct name (or up to 20 characters)	+	+	-	-	+	+	-	-
Entering an incorrect name (or more than 20 characters)	-	-	+	+	-	-	+	+
“Toy Story” movie	+	-	+	-	+	-	+	-
“Spider man” movie	-	+	-	+	-	+	-	+
Actions								
Successful movie booking	+	-	-	-	+	+	-	-
Unsuccessful movie booking	-	+	+	+	-	-	+	+

1. Decision table (to validate all possible test scenarios)

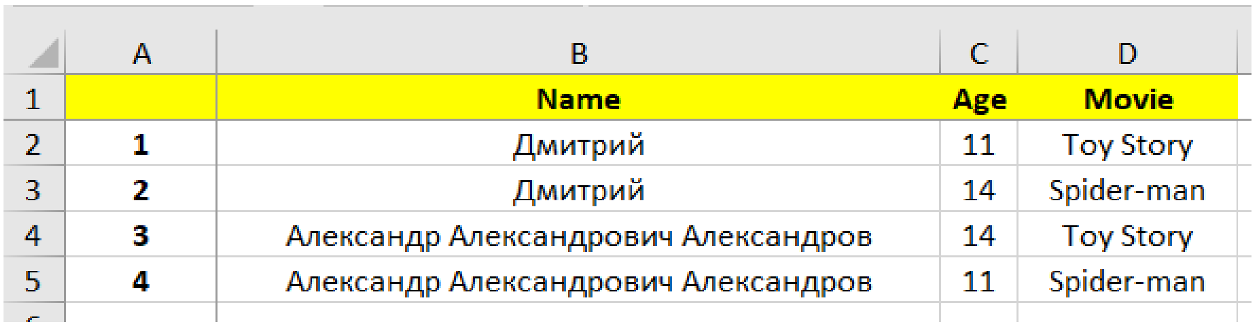
Equivalence classes (this technique is used to break down the input data (into valid and invalid), in this case for age (e.g. age less than 12 years old and 12 years old and above or the number of characters entered before and after 20).

Boundary value analysis (to test on boundary values to ensure the system works correctly on boundary conditions)

Pairwise testing (to reduce test cases)



	Name	Age	Movie
Row 1	Дмитрий	11	Toy Story
Row 2	Александр Александр	14	Spider-man



	A	B	C	D
1		**Name**	**Age**	**Movie**
2	**1**	Дмитрий	11	Toy Story
3	**2**	Дмитрий	14	Spider-man
4	**3**	Александр Александрович Александров	14	Toy Story
5	**4**	Александр Александрович Александров	11	Spider-man

Error anticipation (you can enter parameters where the probability of a defect occurring is high)

2. Jira, YouTrack, Redmine for writing checklists, test cases and bug reports.

JUnit or TestNG for writing and running tests.

Pairwise Tools for applying pairwise testing techniques

Postman, SoapUI, Swagger for API testing (if necessary)

3. Security testing: check for vulnerabilities and the possibility of form hacking.

Compatibility testing: testing how the form works in different browsers and on different devices.

Performance testing: evaluation of form loading and response speed under different loads.

Localization testing: testing how the form works in a different language.

Usability testing: you can add a button to the form to cancel a ticket booking a booking notification or add an email field to receive a link to the ticket.

4. Access to the application source code for deeper analysis.

Operation for canceling a ticket reservation.

Field for entering e-mail.

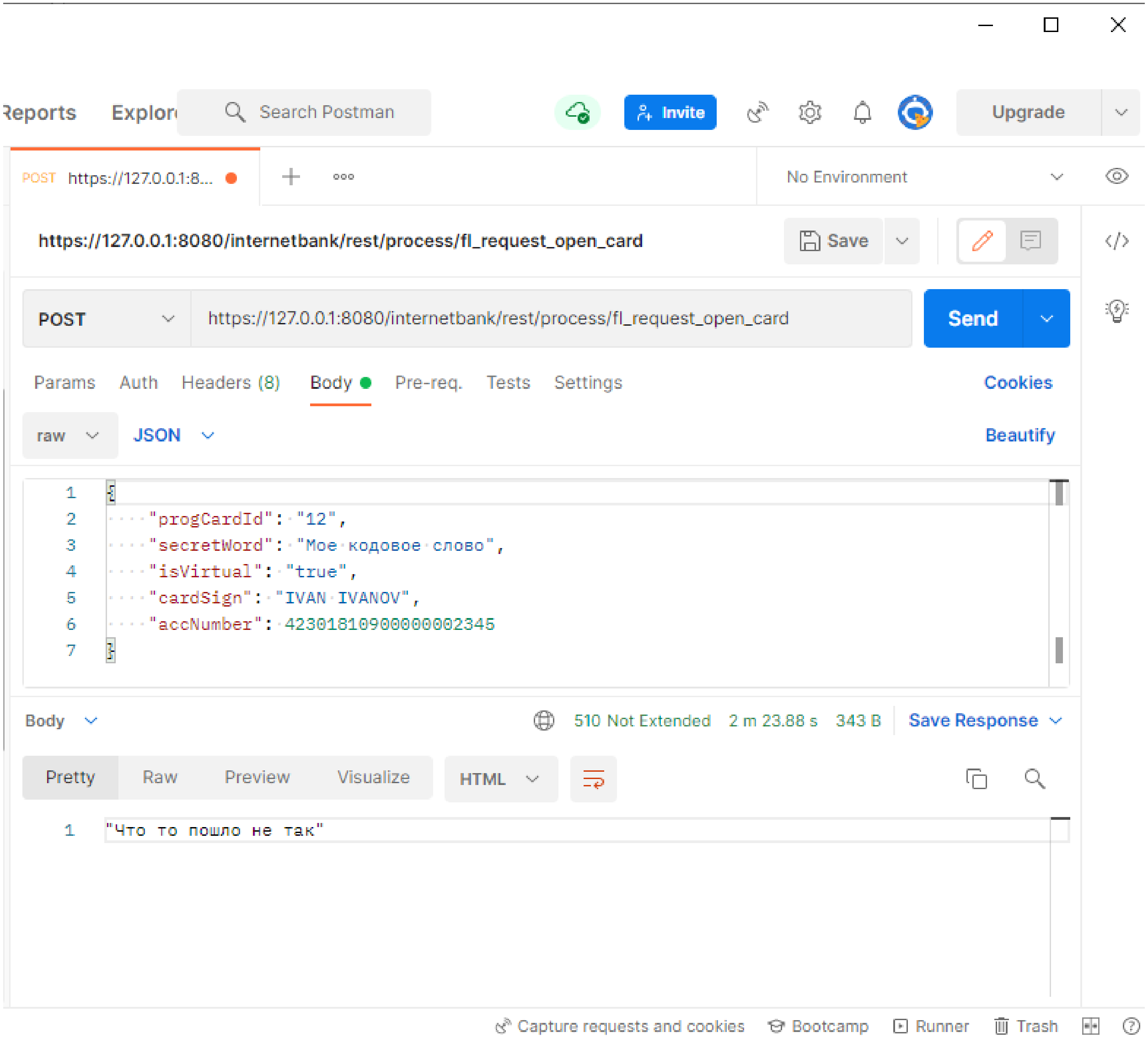
Could I enter characters other than letters in the "name" field?

What is the maximum age allowed and what is the minimum age?

No age restrictions for movies (e.g. 6+ or 12+).

Task №2.

Here is an example of a request error from Postman. This request is sent when an Online Banking user wants to issue a virtual card. Below you will find the documentation for the request. You need to determine what could have gone wrong.



https://127.0.0.1:8080/internetbank/rest/process/fl_request_open_card

POST https://127.0.0.1:8080/internetbank/rest/process/fl_request_open_card

```
{
    "progCardId": "12",
    "secretWord": "Мое кодовое слово",
    "isVirtual": "true",
    "cardSign": "IVAN IVANOV",
    "accNumber": 42301810900000002345
}
```

Body   510 Not Extended   2 m 23.88 s   343 B

HTML

```
"Что то пошло не так"
```

Description of the card issuance application request:

POST request to the server, of the form:

http://127.0.0.1:8080/internetbank/rest/process/fl_request_open_card

Input parameters:

Parameter	Description	Type	Mandatory
progCardId	Id of card product	Long	+
secretWord	Code word	String	-
cardSign	Name on card	String	+
accNumber	Debit account number	Long	+
isVirtual	Sign of virtual card	Boolean	+

Output parameters:
The service should return an applicaton/json response containing the parameters:

Parameter	Description
Id	Identifier of the created application

Answers:

1.HTTP status code 510 may be issued because the server does not have an extension that the client wants to use. The server requires more data in the request.

2.Perhaps not all mandatory parameters were passed in the request. If the "secretWord" parameter is optional, the server may return an error.

3.If the "secretWord" parameter was passed but does not match the expected value, the server may return an error

4.The "accNumber" parameter can contain a number up to 9 223 372 036 854 775 807. The transmitted value exceeds the upper threshold.