МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра информационных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №3**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. А. Иванов

(подпись)

Направление подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем

Направленность Автоматизация тестирования программного обеспечения

Руководитель Н.Ю. Добровольская

(подпись)

Краснодар

2024

Цель: научиться оформлять корректные чек-листы, строить классы эквивалентности и выполнять тестирование граничных значений.

Задание

1. Составить чек-лист для тестирования API добавляющее атрибут пользователю.
2. Разработать набор тест-кейсов, различающихся на одно недопустимое значение. Причём, входные значения (как корректные, так и некорректные) могут варьироваться.
3. Для веб-сайта Регистрационной системы Государственного Университета определить классы эквивалентности и подходящие тест-кейсы.
4. Определить для ипотечной компании “Гуфи” набор тест-кейсов, содержащих комбинации допустимых (на границе) и недопустимых (за границей) значений.
5. Для веб-сайта Регистрационной системы Государственного Университета определить границы и подходящие тестовые граничные значения.

**1 Чек-лист для проверки API**

Вариант чек-листа представлен в виде таблицы 1.

|  |  |  |
| --- | --- | --- |
| Заголовок тест-кейса | Отметки pass/fail | Заметки |
| Проверка добавления атрибута пользователю с корректными данными |  |  |
| Проверка успешного добавления атрибута при передаче правильных параметров запроса (userId, clientType, attributeId, attributeData) |  |  |
| Проверка возврата корректного статуса кода (200 OK) при успешном добавлении атрибута |  |  |
| Проверка корректного формата ответа после добавления атрибута (JSON) |  |  |
| Проверка отсутствия изменений у других пользователей при добавлении атрибута конкретному пользователю |  |  |
| Проверка обработки некорректных данных (неверный тип данных, отсутствие обязательных полей) |  |  |
| Проверка возврата соответствующего статуса кода при ошибке (400 Bad Request) при передаче некорректных данных |  |  |
| Проверка возврата информативного сообщения об ошибке при некорректных данных |  |  |
| Проверка возможности добавления атрибута существующему пользователю |  |  |
| Проверка возможности добавления атрибута новому пользователю |  |  |
| Проверка корректности обработки запроса при отсутствии параметров |  |  |

Таблица 1 – чек-лист для проверки API

**2 Тест-кейсы для ИКГ**

Для того, чтобы составить тест-кейсы, сначала нужно построить корректные классы эквивалентности для тестирования по всем входным данным (рисунки 1 – 4).



Рисунок 1 – класс эквивалентности для дохода (income)

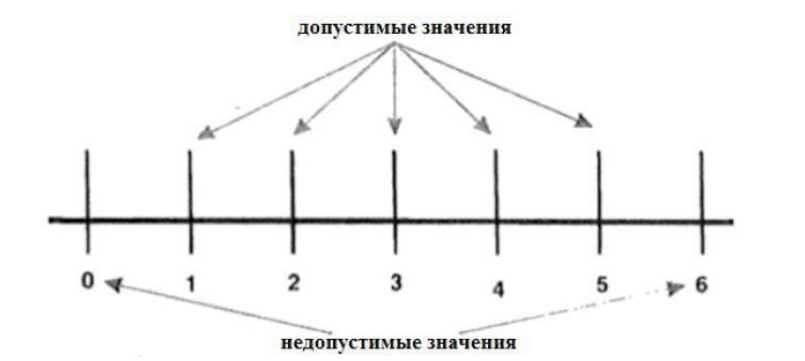


Рисунок 2 – класс эквивалентности для кол-ва домов (houses)



Рисунок 3 – класс эквивалентности для заявителя (applicantType)



Рисунок 4 – класс эквивалентности для собственности (propertyType)

Тест-кейсы для данных классов эквивалентности представлены в таблице 2.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Описание тест-кейса | Входные данные | Ожидаемый результат |
| 1 | Проверка обработки запроса с корректными данными | {"income": 50000, "houses": 3, "applicantType": "человек", "propertyType": "квартира"} | Запрос успешно обработан, выдана ипотека |
| 2 | Проверка обработки запроса с некорректным типом дохода | {"income": "мало", "houses": 2, "applicantType": "человек", "propertyType": "таунхаус"} | Ошибка: "Некорректный тип данных для дохода" |
| 3 | Проверка обработки запроса с недопустимым форматом дохода | {"income": 50000.50, "houses": 3, "applicantType": "человек", "propertyType": "квартира"} | Ошибка: "Некорректный формат дохода" |
| 4 | Проверка обработки запроса с доходом ниже минимального значения | {"income": 500, "houses": 1, "applicantType": "человек", "propertyType": "частный дом"} | Ошибка: "Доход слишком низкий для получения ипотеки" |
| 5 | Проверка обработки запроса с доходом выше максимального значения | {"income": 100000, "houses": 3, "applicantType": "человек", "propertyType": "таунхаус"} | Ошибка: "Доход слишком высокий для получения ипотеки" |
| 6 | Проверка обработки запроса с недопустимым количеством домов | {"income": 60000, "houses": 6, "applicantType": "человек", "propertyType": "частный дом"} | Ошибка: "Недопустимое количество домов для ипотеки" |
| 7 | Проверка обработки запроса с недопустимым типом недвижимости | {"income": 70000, "houses": 2, "applicantType": "человек", "propertyType": "домик на дереве"} | Ошибка: "Недопустимый тип недвижимости для ипотеки" |
| 8 | Проверка обработки запроса от компании (недопустимый тип заявителя) | {"income": 70000, "houses": 3, "applicantType": "компания", "propertyType": "квартира"} | Ошибка: "Ипотека выдается только физическим лицам" |
| 9 | Проверка обработки запроса с пустыми входными данными | {} | Ошибка: "Отсутствуют данные для расчета ипотеки" |

Таблица 2 – тест-кейсы для ИКГ

**3 Классы эквивалентности и тест-кейсы для веб-сайта**

Классы эквивалентности, которые описывают значения всех вводных данных, представлены на рисунках 5 – 10.

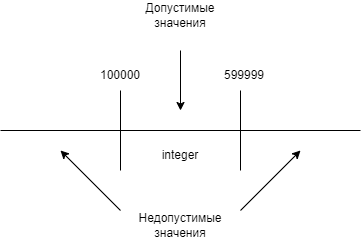


Рисунок 5 – класс эквивалентности для индекса (index)

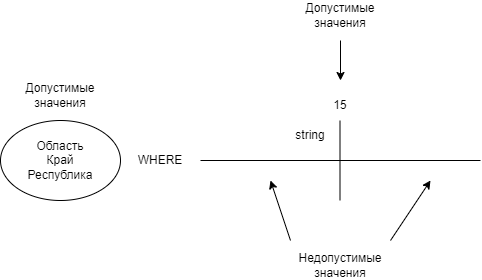


Рисунок 6 – класс эквивалентности для регион (region)

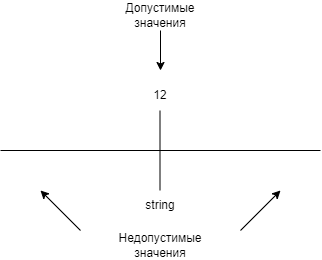


Рисунок 7 – класс эквивалентности для населённого пункта (city)

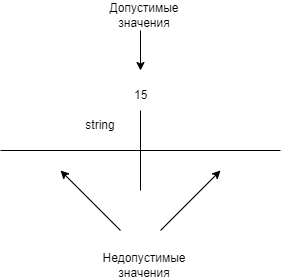


Рисунок 8 – класс эквивалентности для фамилии (surname)

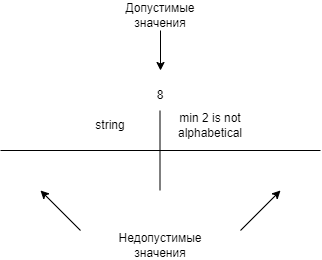


Рисунок 9 – класс эквивалентности для идентификатора пользователя (user\_id)

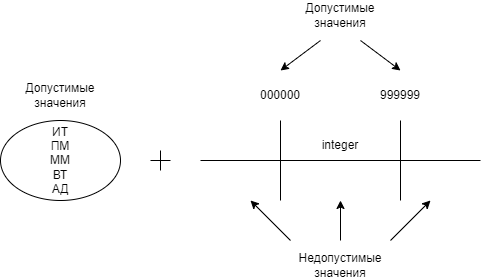


Рисунок 10 – класс эквивалентности для идентификатора студента (student\_id)

Тест-кейсы, удовлетворяющие вышеперечисленным классам эквивалентности, указаны в таблице 3.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Описание тест-кейса | Входные данные | Ожидаемый результат |
| 1 | Проверка на ввод корректных данных | {"index": 350000, "region": "Краснодарский", "city": "Краснодар", "surname": "Иванов", "user\_id": "01abcdef", "student\_id": "ИТ123456"} | Запрос успешно обработан |
| 2 | Проверка на ввод некорректного индекса | {"index": 900560, "region": "Краснодарский", "city": "Краснодар", "surname": "Иванов", "user\_id": "01abcdef", "student\_id": "ИТ123456"} | Ошибка: "Введён некорректный индекс" |
| 3 | Проверка на ввод некорректного региона | {"index": 350000, "region": "Москва", "city": "Краснодар", "surname": "Иванов", "user\_id": "01abcdef", "student\_id": "ИТ123456"} | Ошибка: "Введён некорректный регион" |
| 4 | Проверка на ввод некорректного города | {"index": 350000, "region": "Краснодарский", "city": "Америка", "surname": "Иванов", "user\_id": "01abcdef", "student\_id": "ИТ123456"} | Ошибка: "Введён некорректный город" |
| 5 | Проверка на ввод некорректной фамилии | {"index": 350000, "region": "Краснодарский", "city": "Краснодар", "surname": 123, "user\_id": "01abcdef", "student\_id": "ИТ123456"} | Ошибка: "Введён неверный формат фамилии" |
| 6 | Проверка на ввод некорректного идентификатора пользователя | {"index": 350000, "region": "Краснодарский", "city": "Краснодар", "surname": "Иванов", "user\_id": "#$%", "student\_id": "ИТ123456"} | Ошибка: "Ведён неверный идентификатор пользователя" |
| 7 | Проверка на ввод некорректного идентификатора студента | {"index": 350000, "region": "Краснодарский", "city": "Краснодар", "surname": "Иванов", "user\_id": "01abcdef", "student\_id": "АБ123456"} | Ошибка: "Введена неверная кафедра" |
| 8 | Проверка на ввод пустых данных | {} | Ошибка: "Недостаточно данных для выполнения запроса" |

Таблица 3 – тест-кейсы для веб-сайта