# Test plan

This test plan is divided into 2 tables on 5 pages, containing test suites related to Users and Contacts respectively.

## Users

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **HTTP-method** | **Parameters and fields** | **№** | **Test case** | **Description** | **Type** | **Prio** | **Severity** |
| 1.POST | all fields | 1 | Add new user to the phone book by filling no field | This test case checks, that a user can't be added without filling a field | - | 2 | 2 |
|  |  | 2 | Add new user by filling all fields with unique valid values | This test case checks, that user will be added by filling all fields. | + | 1 | 1 |
|  | id | 3 | Add new user to the phone book by filling only id with a valid value | This test case checks, that user can't be added by filled id | - | 2 | 2 |
|  |  | 4 | Check automatically generated id by adding the user to the phone book without filling id field | This test case checks, that user id can be generated automatically with right value | + | 1 | 1 |
|  |  | 5 | Add new user to the phone book by filling firstName and id with valid value | This test case checks, that a user can be added by filling firstName with valid value and id with integer32,  id=-2147483648, id=2147483647, id with not unique value | + | 1 | 1 |
|  |  | 6 | Add new user to the phone book by filling firstName with valid value and id with invalid value | This test case checks, that user can't be added to the phone book by filling firstName with valid value and id with a space, special characters, letter, numbers and space, fractional numbers, id = -2147483649, id = 2147483648 | - | 2 | 1 |
|  | firstName | 7 | Add new user to the phone book by filling only firstName with a valid value | This test case checks, that a user can be added by filling only firstName field with 2 letters, with 15 letters, uppercase and lowercase letters, numbers, special characters, space between them | + | 1 | 1 |
|  |  | 8 | Add new user to the phone book by filling only firstName with invalid value | This test case checks, that a user can't be added by filling only firstName field with 1 letter, 16 letters, with two spaces | - | 2 | 2 |
|  | lastName | 9 | Add new user to the phone book by filling only last name with valid value | This test case checks, that a user can't be added by filling only lastName field | - | 2 | 2 |
|  |  | 10 | Add new user to the phone book by filling firstName and lastName with a valid value. | This test case checks, that a user can be added by filling firstName and lastName with a valid value (uppercase and lowercase letters, numbers, special characters, space between them, 1 character) | + | 1 | 1 |
| 2.GET | no parameters | 11 | Search for users by filling no searh criteria | This test case checks, that list of all users can be returned by filling no searh criteria | + | 1 | 1 |
|  | firstName | 12 | Search for users by typing no character in the parameter name | This test case checks, that user can't be returned by typing no character in the parameter name | - | 2 | 3 |
|  |  | 13 | Search for a user by typing a valid firstName | This test case checks, that user can be returned by typing firstName, only part of a firstName, firstName and space after it, not unique firstName, firstName with a lowercase letter, firstName and a special character after it, firstName with a special character between letters in parameter name | + | 1 | 1 |
|  |  | 14 | Search for a user by typing invalid firstName in parameter name | This test case checks, that a user can't be returned by filling no firstName, by filling firstName with a number, space, special character, valid firstName and a number, firstName and a special character, firstName with a space between letters in parameter name | - | 2 | 2 |
|  |  | 15 | Search for a deleted or never existed user by typing the firstName in parameter name | This test case checks, that deleted or never existed user will not be found | - | 2 | 3 |
|  | user id | 16 | Search for a user by typing a valid id | This test case checks, that a user can be returned by typing its id, only a part of id, id and a space | + | 1 | 1 |
|  |  | 17 | Search for a user by typing an invalid id | This test case checks, that a user can't be returned if the user has never existed, was deleted, by typing id and a space, id and a special character, space, empty id | - | 2 | 3 |
| 3.DELETE | id | 18 | Delete a user typing its valid id | This test case checks, that a user can be deleted by typing its id | + | 1 | 1 |
|  |  | 19 | Delete a user typing invalid id | This test case checks, that a user can't be deleted by typing no id, id=0, its id and a space, its id and a special character, a zero before id | - | 2 | 3 |
| 4.PUT | Change user firstName | 20 | User's firstName can't be changed by typing invalid user id and a new valid firstName | This test case checks, that user's firstName can't be changed by typing no id, id=0, id and a space, id and a special character, a letter, a zero before id, deleted or not existing id + a new valid firstName | - | 2 | 2 |
|  |  | 21 | User's firstName can't be changed by typing an existing user id and invalid firstName | This test case checks, that user's firstName can't be changed by typing valid user id + invalid value of firstName like a space, a special character, empty firstName, consisting 1 or 16 characters, | - | 2 | 3 |
|  |  | 22 | User's firstName can be changed by typing an existing user id and a new valid firstName | This test case checks, that user's firstName can be changed by typing an existing user id + a new valid firstName consisting of 2, 15 characters, containing a space, a number, a special character | + | 1 | 1 |
|  | Change user lastName | 23 | User's lastName can be changed by typing an existing user id, appropriate firstName and a new valid lastName | This test case checks, that user's lastName can be changed by typing an existing user id, appropriate firstName and a valid lastName containing a number, uppercase and lowercase letters, special characters, a space | + | 1 | 1 |
|  |  | 24 | User's lastName can't be changed by typing an existing user id, appropriate firstName and an invalid lastName | This test case checks, that user's lastName can't be changed by typing an existing user id, appropriate firstName and a lastName consisting of a space or a special character | - | 2 | 3 |

## Contacts

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **HTTP-method** | **Parameters and fields** | **№** | **Test case** | **Description** | **Type** | **Prio** | **Severity** |
| 1. POST | all fields | 1 | Add new contact to the phone book by filling no field | This test case checks, that a contact can't be added without filling a field | - | 2 | 3 |
|  |  | 2 | Add new contact by filling all fields with unique valid values | This test case checks, that contact will be added by filling all fields. | + | 1 | 1 |
|  | id | 3 | Add new contact by filling only id with a valid value | This test case checks, that contact can't be added by filled id | - | 2 | 2 |
|  |  | 4 | Check automatically generated id by adding the first contact | This test case checks, that contact id can be generated automatically with right value | + | 1 | 2 |
|  |  | 5 | Add new contact by filling phone with valid value and id with valid value | This test case checks, that a contact can be added by filling phone with valid value and id with integer32, id = -2147483648, id = 2147483647, id with not unique value | + | 1 | 1 |
|  |  | 6 | Add new contact by filling phone with valid value and id with invalid value | This test case checks, that contact can't be added to the phone book by filling phone with valid value and id with a space, special characters, letter, numbers and space, fractional numbers, id = -2147483649, id = 2147483648, id with a numbers and a space between them | - | 2 | 3 |
|  | firstName | 7 | Add new contact by filling phone and firstName with valid values | This test case checks, that contact can be added with filling phone and firstName with not unique valid value, firstName with 1 letter, firstName with a space, firstName with uppercase and lowercase letters, numbers, special characters, space between them | + | 1 | 1 |
|  |  | 8 | Add new contact by filling only firstName with unique valid value | This test case checks, that contact can't be added by filled only first name | - | 2 | 3 |
|  | lastName | 9 | Add new contact by filling phone and lastName with valid values | This test case checks, that contact can be added with filling phone and lastName with not unique valid value, lastName with 1 letter, lastName with a space, lastName with uppercase and lowercase letters, numbers, special characters, space between them | + | 1 | 1 |
|  |  | 10 | Add new contact by filling only lastName with unique valid value | This test case checks, that contact can't be added by filled only lastName | - | 2 | 3 |
|  | phone | 11 | Add new contact by filling phone with a valid value | This test case checks, that contact with valid phone contains 10 digits, 9 digits, not unique value can be added to phone book | + | 1 | 1 |
|  |  | 12 | Add new contact by filling phone with invalid value | This test case checks, that contact with phone contains 11 digits, phone with letters, numbers, special characters, 10 zeros and space between symbols can't be added to phone book | - | 2 | 3 |
|  | email | 13 | Add new contact by filling only email with a unique valid value | This test case checks, that contact can't be added by filled only email | - | 2 | 3 |
|  |  | 14 | Add new contact by filling phone and email with a valid values | This test case checks, that contact can be added with filling phone and email with valid value containing character @, upper case and lowercase letters, numbers, special characters, | + | 1 | 1 |
|  |  | 15 | Add new contact by filling phone and email with invalid values | This test case checks, that contact can't be added by filled phone with valid value and email containing space between letters, only one letter, and no character @ | - | 2 | 2 |
| 2. GET | no field | 16 | Search for contacts by filling no searh criteria | This test case checks, that no contact will be found when no search criteria or user is given | - | 2 | 2 |
|  | user id | 17 | Search for all the contacts of the existing user | This test case checks, that all contacts of appropriate user will be returned, when user id is given | + | 1 | 1 |
|  |  | 18 | Search for all the contacts of the deleted or not existing user | This test case checks, that no contact of deleted or not existing user can be returned | - | 2 | 2 |
|  |  | 19 | Search for all the contacts by typing invalid user id | This test case checks, that no contact can be returned by typing invalid user id (containing a special character, a space) | - | 2 | 2 |
|  | phone | 20 | Search for a contact of the existing user by existing phone | This test case checks, that appropriate contact can be returned by typing valid existing user id and phone | + | 1 | 1 |
|  |  | 21 | Search for a contact of the existing user by number beginning with a zero | This test case checks, that appropriate contact can't be returned by typing existing user id and invalid phone (consisting of 9 or 11characters, beginning with a zero, containing 10 zeros, only a part of a number, letter, special character or a space, deleted or not existing phone) | - | 2 | 2 |
|  | contact id | 22 | Search for a contact of the existing user by invalid contact id | This test case checks, that appropriate contact can't be returned by invalid contact id (id of deleted or not existing contact, id and a space, id and a special character, id and a letter, empty id, a space, a special character, a letter, id=0, id beginning with a zero) | - | 2 | 2 |
|  | other | 23 | Search for a contact of the existing user by email | This test case checks, that appropriate contact can't be returned by typing an email | - | 3 | 3 |
|  |  | 24 | Search for a contact of the existing user by contact firstName | This test case checks, that appropriate contact can't be returned by typing contact firstName | - | 3 | 3 |
|  |  | 25 | Search for a contact of the existing user by contact lastName | This test case checks, that appropriate contact can't be returned by typing contact lastName | - | 3 | 3 |
| 3. DELETE | user Id | 26 | Delete a contact by typing no user id | This test case checks, that appropriate contact can't be deleted by typing no user id | - | 2 | 2 |
|  |  | 27 | Delete a contact by typing invalid user id | This test case checks, that appropriate contact can't be deleted by typing invalid user id (id of deleted or not existing contact, id and a space, id and a special character, id and a letter, empty id, a space, a special character, a letter, id=0, id beginning with a zero) | - | 2 | 3 |
|  | contact id | 28 | Delete a contact by typing its user and its contact id | This test case checks, that appropriate contact can be deleted by typing valid user id and valid contact id | + | 1 | 1 |
| 4. PUT |  | 29 | Change a value of 1 optional field by typing a valid existing id and a valid existing phone | This test case checks, that all contact fields can be edited at once by typing valid user id and contact id and new valid values | + | 1 | 1 |
|  |  | 30 | Change a value of all optional fields by typing a valid existing id and a valid existing phone | This test case checks, that appropriate contact field can be edited by typing valid user id and contact id and new valid values | + | 1 | 1 |
|  |  | 31 | Change a value of 1 optional field by typing invalid user id and a valid existing phone | This test case checks, that appropriate contact field can't be edited by typing invalid user id, valid contact id, phone and new valid value | - | 2 | 2 |
|  |  | 32 | Change a value of 1 optional field by typing valid user id, invalid contact id and a valid existing phone | This test case checks, that appropriate contact field can't be edited by typing valid user id, invalid contact id, phone and new valid value | - | 2 | 2 |
|  | phone | 33 | Change a value of 1 optional field by typing an existing id and invalid or empty phone | This test case checks, that appropriate contact field can't be edited by typing valid user id, valid contact id, invalid phone and new valid value | - | 2 | 2 |
|  |  | 34 | Change a value of 1 optional field to invalid by typing a valid existing id and a valid existing phone | This test case checks, that appropriate contact field can't be edited by typing valid user id, valid contact id, valid phone and invalid new value of a parameter to be updated | - | 2 | 2 |