1. Introduction

Referring to Flasky documentation, we have:

Acceptance Criteria / EPICs

As a UI user I can:

1: Register through web portal

2: Review my own user information from the main view

As an API Consumer I can:

1: Review users registered in system

2: If authenticated I can get personal information of users

3: If authenticated I can update personal information of users

In this document, we will break down UI EPIC #1 and API Epic #3 into user stories/requirements.

While doing so, we will also provide flowcharts on

- how registration form works for UI EPIC #1
- How API works for API Epic #3

2. EPIC: As a UI user, I can register through web portal

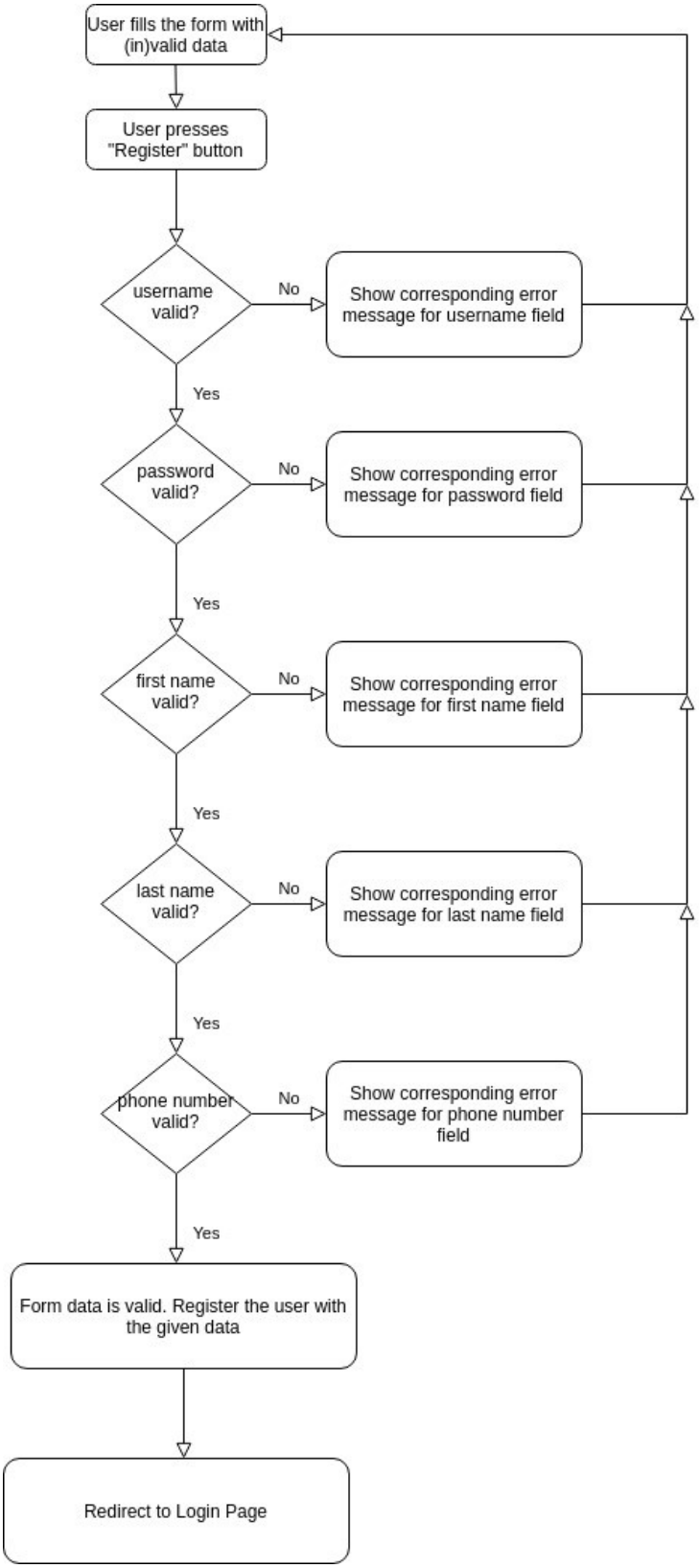

Figure 1 – In case of errors, registration form gives 1 error at a time from top to bottom

As shown in **Figure 1**, the design of the registration form has chosen a lean approach; Instead of showing all the errors at once, it shows the errors 1 error at a time from top to bottom. So, each time user corrects the error and presses “Register” button, then the form shows the next error and so on until there are no more errors in the form data. Then registration must succeed and user must be redirected to login page.

1) All 5 fields are mandatory; If any of the fields is missing, then "Please fill out this field." error message is shown. Note that **Figure 1** still applies; if many fields are missing, then only the topmost one’s error message is shown.

3) All 5 fields are verified against their respective requirements:
a) Username field’s rules:

Requirement ID / Tag	**Description**	**Status**
REQ-REG-FORM-USERNAME-1	Referring to Fig-1, among other fields, username field has the first priority when it comes to showing error	FAIL
REQ-REG-FORM-USERNAME-2	Minimum 8 characters	FAIL
REQ-REG-FORM-USERNAME-3	Alpha numeric characters and non-alpha numeric characters and their mix are allowed	PASS
REQ-REG-FORM-USERNAME-4	Space character is not allowed	FAIL
REQ-REG-FORM-USERNAME-5	If username is missing, then "Please fill out this field." error message is shown. Note that **Figure 1** still applies; if many fields are missing or incorrect, then username’s error message is shown.	PASS

Table 1

b) Password field’s rules

Requirement ID / Tag	**Description**	**Status**
REQ-REG-FORM-PASSWORD-1	Referring to Fig-1, among other fields, password field has the second priority when it comes to showing error	FAIL
REQ-REG-FORM-PASSWORD-2	Minimum 8 characters	FAIL
REQ-REG-FORM-PASSWORD-3	Must include at least one character from the sets [a-z], [A-Z], [0-9] and [!?.]	FAIL
REQ-REG-FORM-PASSWORD-4	If password is missing, then "Please fill out this field." error message is shown. Note that **Figure 1** still applies; if many fields are missing or incorrect, then only the topmost one’s error message is shown	FAIL

Table 2

c) First Name field’s rules:

Imagine there is a person with full name: Helena Margaretha Holm Cuzdan.
Note that Margaretha is the 2.nd first name and Cuzdan is the 2.nd last name.

Imagine there is a person with full name: Ha Xi (a valid Vietnamese full name)

Requirement ID / Tag	Description	Status
REQ-REG-FORM-FIRST-NAME-1	Referring to Fig-1, among other fields, first name field has the third priority when it comes to showing error	FAIL
REQ-REG-FORM-FIRST-NAME-2	Minimum 2 characters for each word (e.g. Ha Xi but not H X, or Ha but not H)	FAIL
REQ-REG-FORM-FIRST-NAME-3	Must contain letters from the set [a-zA-Z] for each word	PASS
REQ-REG-FORM-FIRST-NAME-5	Cannot contain numbers	FAIL
REQ-REG-FORM-FIRST-NAME-6	Cannot contain non-alphanumeric characters	FAIL
REQ-REG-FORM-FIRST-NAME-7	Can contain two first names, which must be separated with a single space (e.g. 'Holm Cuzdan' but not 'Holm Cuzdan')	FAIL
REQ-REG-FORM-FIRST-NAME-8	If first name is missing, then "Please fill out this field." error message is shown. Note that **Figure 1** still applies; if many fields are missing or incorrect, then only the topmost one’s error message is shown	FAIL

Table 3

d) Last Name

Requirement ID	Description	Status
REQ-REG-FORM-LAST-NAME-1	Referring to Fig-1, among other fields, last name field has the fourth priority when it comes to showing error	FAIL
REQ-REG-FORM-LAST-NAME-2	Each word must be at least 2 letters (e.g. Ha Xi but not H X, or Ha but not H)	FAIL
REQ-REG-FORM-LAST-NAME-3	Must contain letters from the set [a-zA-Z] for each word	PASS
REQ-REG-FORM-LAST-NAME-5	Cannot contain numbers	FAIL
REQ-REG-FORM-LAST-NAME-6	Cannot contain non-alphanumeric characters	FAIL
REQ-REG-FORM-LAST-NAME-7	A single space must separate the last names, if there are 2 last names (e.g. 'Holm Cuzdan' but not 'Holm Cuzdan')	FAIL
REQ-REG-FORM-LAST-NAME-8	If last name is missing, then "Please fill out this field." error message is shown. Note that **Figure 1** still applies; if many fields are missing or incorrect, then only the topmost one’s error message is shown	FAIL

Table 4

e) Phone number

This is a mobile phone number from any country, not a fixed line. Referring to https://en.wikipedia.org/wiki/List_of_mobile_telephone_prefixes_by_country

Requirement ID/Tag	Description	Status
REQ-REG-FORM-PHONE-NUMBER-1	May contain a single '+' character as the first character and the rest must be numbers	FAIL
REQ-REG-FORM-PHONE-NUMBER-2	If REQ-REG-FORM-PHONE-NUMBER-1 is not met, then it must contain only numbers	PASS
REQ-REG-FORM-PHONE-NUMBER-3	Must not contain alphabetic characters [a-zA-Z]	FAIL
REQ-REG-FORM-PHONE-NUMBER-4	Must not contain non-alphanumeric characters	FAIL
REQ-REG-FORM-PHONE-NUMBER-5	Must not contain space(s)	FAIL
REQ-REG-FORM-PHONE-NUMBER-6	Referring to Fig-1, among other fields, phone number field has the least priority when it comes to showing error for a typed phone number	FAIL
REQ-REG-FORM-PHONE-NUMBER-7	If phone number is missing, then "Please fill out this field." error message is shown. Referring to Fig-1, among other fields, phone number field has the least priority when it comes to showing error	FAIL

Table 5

IMPORTANT! Each requirement may have multiple test cases. Requirement Ids are used as tags in automated test cases.

3. EPIC: As an API User, If authenticated I can update personal information of users

In the next page, we give how api is supposed to work when it comes to the epic in question.

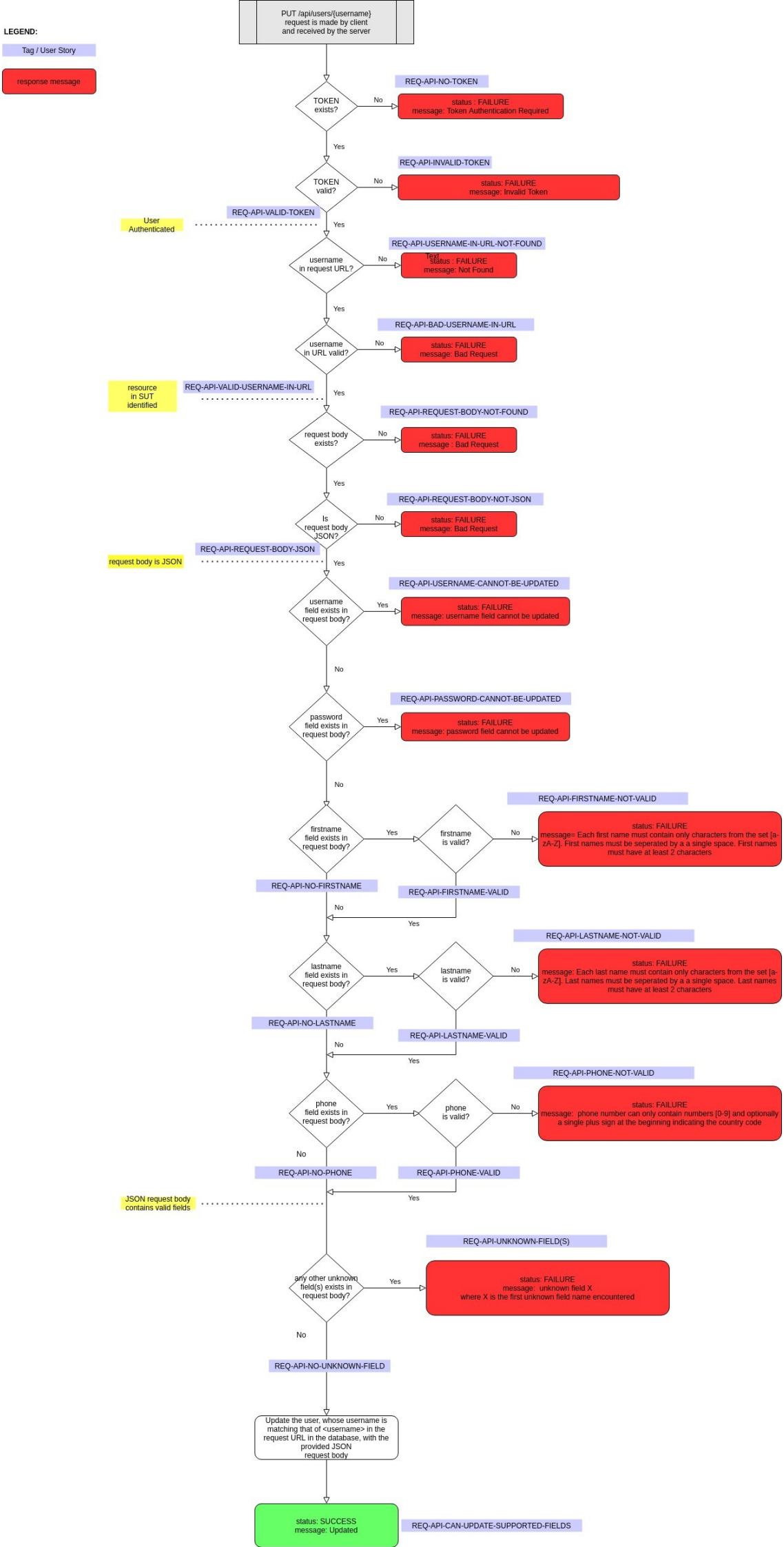
LEGEND:
Tag / User Story
response message
PUT /api/users/{username}
request is made by client
and received by the server
TOKEN
exists?
No
REQ-API-NO-TOKEN
status : FAILURE
message: Token Authentication Required
Yes
TOKEN
valid?
No
REQ-API-INVALID-TOKEN
status: FAILURE
message: Invalid Token
REQ-API-VALID-TOKEN
User
Authenticated
Yes
username
in request URL?
No
REQ-API-USERNAME-IN-URL-NOT-FOUND
status : FAILURE
message: Not Found
Yes
username
in URL valid?
No
REQ-API-BAD-USERNAME-IN-URL
status: FAILURE
message: Bad Request
resource
in SUT
identified
REQ-API-VALID-USERNAME-IN-URL
Yes
request body
exists?
No
REQ-API-REQUEST-BODY-NOT-FOUND
status: FAILURE
message : Bad Request
Yes
Is
request body
JSON?
No
REQ-API-REQUEST-BODY-NOT-JSON
status: FAILURE
message: Bad Request
request body is JSON
REQ-API-REQUEST-BODY-JSON
Yes
username
field exists in
request body?
Yes
REQ-API-USERNAME-CANNOT-BE-UPDATED
status: FAILURE
message: username field cannot be updated
No
password
field exists in
request body?
Yes
REQ-API-PASSWORD-CANNOT-BE-UPDATED
status: FAILURE
message: password field cannot be updated
No
firstname
field exists in
request body?
Yes
firstname
is valid?
No
REQ-API-FIRSTNAME-NOT-VALID
status: FAILURE
message= Each first name must contain only characters from the set [a-zA-Z]. First names must be seperated by a a single space. First names must have at least 2 characters
REQ-API-NO-FIRSTNAME
REQ-API-FIRSTNAME-VALID
No
Yes
lastname
field exists in
request body?
Yes
lastname
is valid?
No
REQ-API-LASTNAME-NOT-VALID
status: FAILURE
message: Each last name must contain only characters from the set [a-zA-Z]. Last names must be seperated by a a single space. Last names must have at least 2 characters
REQ-API-NO-LASTNAME
REQ-API-LASTNAME-VALID
No
Yes
phone
field exists in
request body?
Yes
phone
is valid?
No
REQ-API-PHONE-NOT-VALID
status: FAILURE
message: phone number can only contain numbers [0-9] and optionally a single plus sign at the beginning indicating the country code
No
REQ-API-NO-PHONE
REQ-API-PHONE-VALID
Yes
JSON request body
contains valid fields
any other unknown
field(s) exists in
request body?
Yes
REQ-API-UNKNOWN-FIELD(S)
status: FAILURE
message: unknown field X
where X is the first unknown field name encountered
No
REQ-API-NO-UNKNOWN-FIELD
Update the user, whose username is matching that of <username> in the request URL in the database, with the provided JSON
request body
status: SUCCESS
message: Updated
REQ-API-CAN-UPDATE-SUPPORTED-FIELDS

Figure 2 – How API works when it comes to the epic “Updating personal information of users”

Referring to Fig. 2, we have the following user stories & their status:

(*) Exploratory tested with POSTMAN only

(**) Automated & tagged

Definitions:

(1) System User: The user, who is registered to the system under test with a valid token

(2) API User: The user, who calls the API to update personal information of the system user

Tag / User Story	**Description**	**Status**	**Explanation**
REQ-API-NO-TOKEN	When no token is provided with otherwise valid update request	PASSED (*)	
REQ-API-INVALID-TOKEN	When the given token is invalid with otherwise valid update request	PASSED (*)	
REQ-API-VALID-TOKEN	When the given token is valid with a (in)valid update request. This is an epic containing the sub user stories in the subsequent rows.	FAILED (*)	Some of the subsequent sub stories failed
REQ-API-USERNAME-IN-URL-NOT-FOUND	When the token is valid, we make a PUT request to /api/users with otherwise valid update request	FAILED (*)	API returns HTML, when JSON is expected
REQ-API-BAD-USERNAME-IN-URL	We make a PUT request to /api/users/non-existing-user with otherwise valid update request	FAILED (*)	500 Internal server error
REQ-API-VALID-USERNAME-IN-URL	When the given username is valid with a (in)valid update request. This is an epic containing the sub user stories in the subsequent rows.	FAILED (*)	Some of the subsequent sub stories failed
REQ-API-REQUEST-BODY-NOT-FOUND	For otherwise a valid request, we provide no body in request	FAILED (*)	API returns HTML, when JSON is expected
REQ-API-REQUEST-BODY-NOT-JSON	For otherwise a valid request, we provide XML/HTML body in request	FAILED (*)	API returns HTML, when JSON is expected
REQ-API-REQUEST-BODY-JSON	When the given request body is JSON with a (in)valid update request. This is an epic containing the sub user stories in the subsequent rows.	FAILED (*)	Some of the subsequent sub stories failed
REQ-API-USERNAME-CANNOT-BE-UPDATED	Using all valid tokens, each time we attempt to update **username** of all users with valid <username> in request URL and a JSON request body	FAILED (**)	Refer to the robot’s test report. API returns HTML

			response, but JSON is expected
REQ-API-PASSWORD-CANNOT-BE-UPDATED	Using all valid tokens, each time we attempt to update **password** of all users with valid <username> in request URL and a JSON request body	FAILED (**)	Refer to the robot's test report. API returns HTML response, but JSON is expected
REQ-API-FIRSTNAME-NOT-VALID	Using all valid tokens, each time we attempt to update **firstname** of all users with valid <username> in request URL and a JSON request body **but with invalid firstname data**	FAILED (**)	Refer to the robot's test report. API returns HTML response, but JSON is expected
REQ-API-FIRSTNAME-VALID	Using all valid tokens, each time we attempt to update **firstname** of all users with valid <username> in request URL and a JSON request body and **with valid firstname data**	FAILED (**)	Refer to the robot's test report. API returns HTML response, but JSON is expected
REQ-API-NO-FIRSTNAME	Using all valid tokens, each time we attempt to update **lastname/phone** of all users with valid <username> in request URL and a JSON request body and **with (in)valid lastname/phone data.** This is an epic containing the sub user stories in the subsequent rows.	FAILED (**)	Refer to the robot's test report. API returns HTML response, but JSON is expected
REQ-API-LASTNAME-NOT-VALID	Using all valid tokens, each time we attempt to update **lastname** of all users with valid <username> in request URL and a JSON request body **but with invalid lastname data**	FAILED (**)	Refer to the robot's test report. API returns HTML response, but JSON is expected
REQ-API-LASTNAME-VALID	Using all valid tokens, each time we attempt to update **lastname** of all users with valid <username> in request URL and a JSON request body and **with valid lastname data**	FAILED (**)	Refer to the robot's test report. API returns HTML response, but JSON is expected
REQ-API-NO-LASTNAME	Using all valid tokens, each time we attempt to update **phone** of all users with valid <username> in request URL and a JSON request body and **with**	FAILED (**)	Refer to the robot's test report. API returns HTML

	(in)valid phone data. This is an epic containing the sub user stories in the subsequent rows.		response, but JSON is expected
REQ-API-PHONE-NOT-VALID	Using all valid tokens, each time we attempt to update **phone** of all users with valid <username> in request URL and a JSON request body **but with invalid phone data**	FAILED (**)	Refer to the robot's test report. API returns HTML response, but JSON is expected
REQ-API-PHONE-VALID	Using all valid tokens, each time we attempt to update **phone** of all users with valid <username> in request URL and a JSON request body and **with valid phone data**	FAILED (**)	Refer to the robot's test report. API returns HTML response, but JSON is expected
REQ-API-NO-PHONE	We have a mix of valid firstname & lastname in request body. Using all valid tokens, each time we attempt to update **those field(s)** of all users with valid <username> in request URL and a JSON request body**.**	FAILED	Refer to the robot's test report. API returns HTML response, but JSON is expected
REQ-API-UNKNOWN-FIELD(S)	In request body, we have field(s) that are unknown to system under test (SUT). Otherwise, we have all the requirements for the request satisfied, such as token, request URL, JSON request body and known fields in JSON like firstname, lastname and phone.	FAILED	Refer to the robot's test report. API returns HTML response, but JSON is expected

Table 6

IMPORTANT: You can refer to the Robot's test report, where test cases are properly tagged with the tags referred in the above table. Also, you can refer to **TestProject/Postman/CRF Demo App & API.postman_collection.json** file having the POSTMAN call collection for exploratory testing.