TEST REPORT

GoRest API

Created by: Zuzanna Pilat

Contents

1.   Introduction

- The purpose of the document is to describe the activities performed in the testing of the GoRest API and to provide an overall assessment of the quality of the software under test.

2.Range and results of tests

2.1 Testing process

Testing of the application took place according to the schedule included in the test plan. Manual tests included:

- Basic CRUD flows (Create, Read, Update, Delete) for users.

- Positive (happy-path) and negative input validation checks.

- Response codes, response body content, and minimal pagination checks.

2.2 Range and status of the tests

| Module name	Description	Status	Comment
Registration	Registration of a basic user account with an e-mail address	POSITIVE	Registering the account is possible
Retrieve data	Retrieve the data of the created user	NEGATIVE	Retrieval impossible
Updating data	Updating created account with new data	POSITIVE	Updating possible
Deleting data	Deleting created user	NEGATIVE	Deleting impossible

2.2.1. Out of scope

- Automation (scripts, CI),

- Performance/load testing

- Deep security testing

- Database/internal checks.

2.3. Metrics

Defects were detected during the tests, the data is presented in tables and graphs.

Data in this part of the report is simulated for demonstration

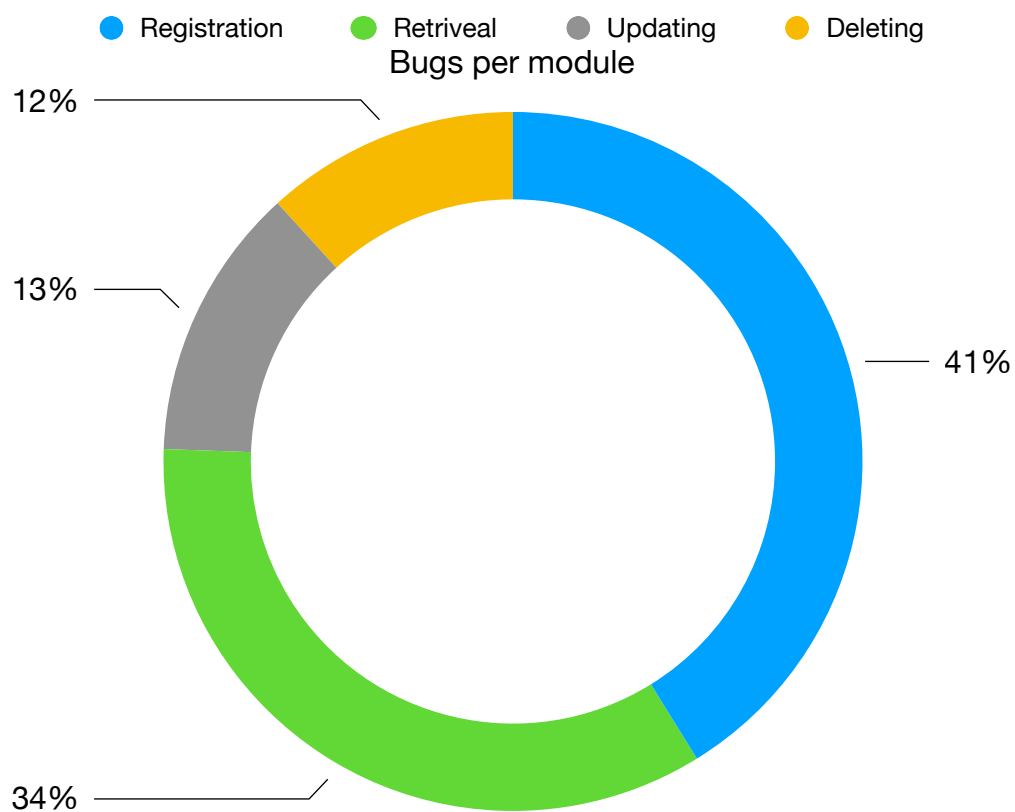| Planned tests	Performed tests	Positive	Negative
62	62	30	32

Positive ●   Negative ●

Test cases

38%

63%
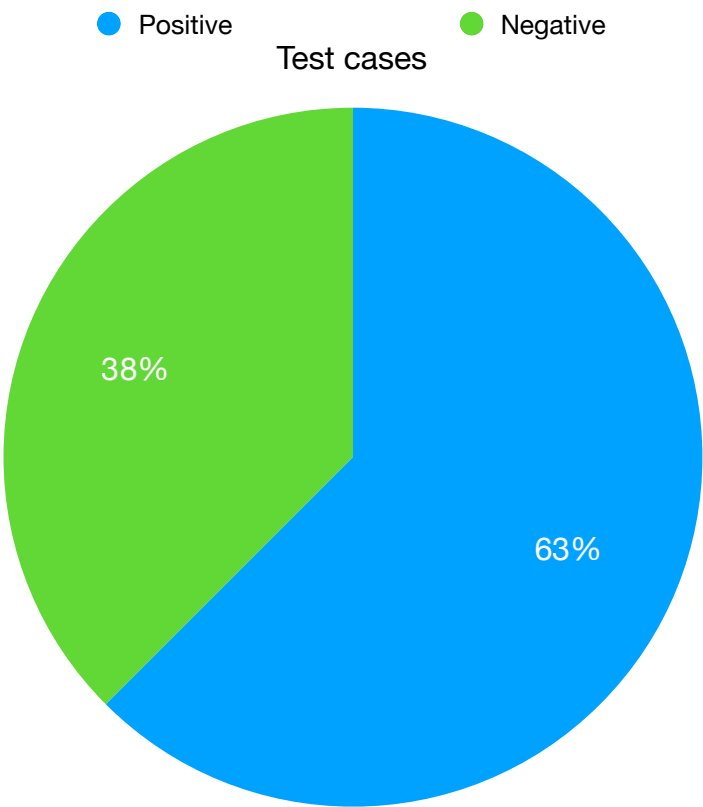
2.3.2. Defects found - priorities and statuses

Status	Critical	High	Medium	Trivial	Sum
Closed	8	8	2	0	18
Open	2	6	4	2	14
					32

Defects Found
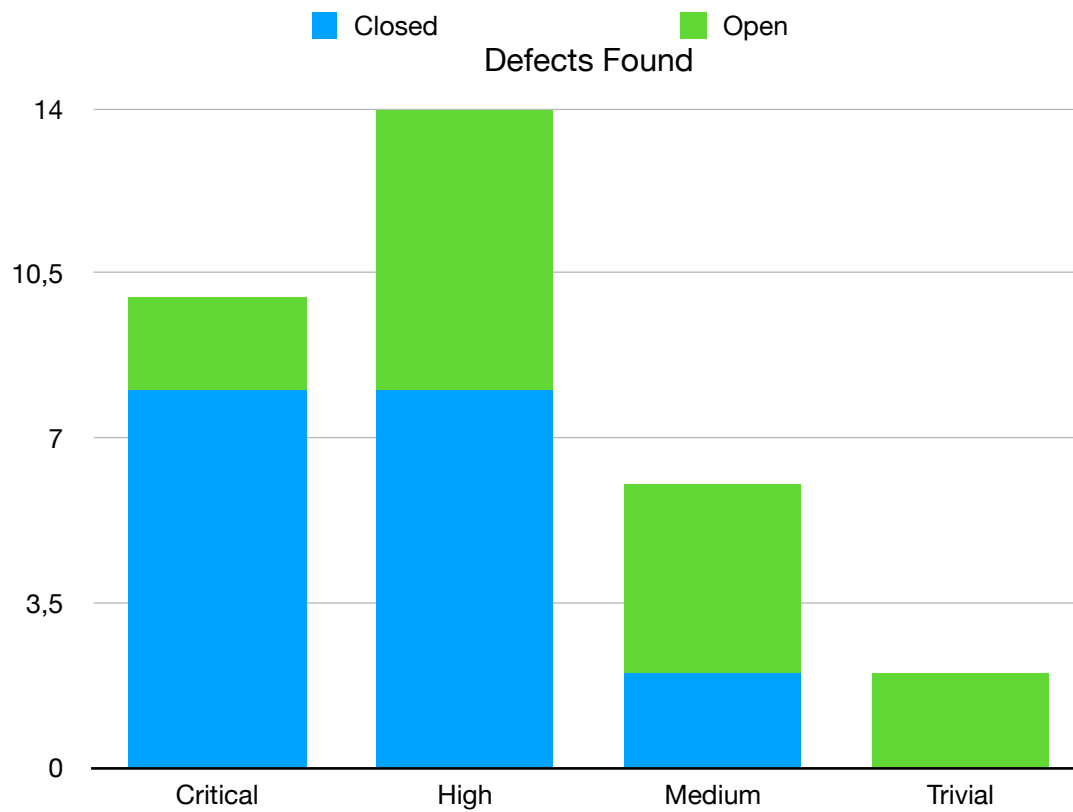
2.3.3. Distribution of bugs per module - basic

Registration	Retrieval	Updating	Deleting	Total
4	14	12	2	32

Bugs per module

2.4. Environment

Environment	Test
System	macOS Monterey 12.7.6
Device	Mac Book Air (13-inch, 2017)
Software	Postman

3. Summary

API status: NEGATIVE

Performed tests did not define the API status as good.

No blocking defects were detected during the tests, yet the critical and important defects detected are still to be fixed. Therefore, the quality of the API cannot be described as good. Due to the existing bugs, it is recommended to fix them.

The defects found should be fixed and then tested. This should be followed by regression testing again to ensure that the corrections made have not introduced other defects.