Test plan

S3-ITS

Author: Tsvetislav Rangelov S3-CB01

Introduction

This document describes my testing strategy, testing workflow and indicates specifics surrounding different types of tests performed in the project.

Test strategy

Albeit the project being developed only by me, it is of utmost importance that testing is handled properly and without compromise. That is why 3 types of tests will be performed, namely – Integration tests, Unit tests and finally, User Acceptance tests. Of course, the test strategy can be extended with more types of tests, but regarding the scope of the project, the aforementioned types are sufficient.

User Acceptance Tests

User Acceptance tests are arguably the least technical and will be used to extract valuable feedback from the future users of this application and stakeholders as well. The main focus would be to ensure that the application workflow goes through smoothly while the user is interacting with it and mainly to cater to their preferences for user interface features.

Unit tests

Unit tests are at the heart of every professional software implementation, with this project not being excluded. The main goal of the unit tests would be to test the functionality of the software business layer and where possible also test for caught exceptions at runtime.

An important note to include would be that the business layer will be tested against the persistence layer in real-time utilizing a containerized Docker database. Essentially, that containerized database would be fired up every time before the tests actually go through so that a legitimate mock connection between the business and persistence is established. The benefit of this approach is that the tests get as close to the production flow of the program as possible. Additionally, if some major changes take place in either of these layers and something breaks unexpectedly, this can be easily handled due to some tests failing/not being ran at all.

Integrations tests

Lastly, Integration tests can provide the testing of the coupled business layer and persistence layer, just like in the unit tests. Difference is that while testing both layers concurrently, a synchronic approach still takes place, in that the most low-level modules are being tested first, followed by any higher level modules that can have dependencies linked together. This is referred to as bottom-up testing and is especially useful to follow the process of how the software components link to each other.

User acceptance tests

US	Name	Steps	Test data
US1	Log in	1.Navigate to page 2.Enter test data in fields 3.Press log in button	***LogIn data:*** *Email: "testmail@gmail.com"* *Password: "12345"*
US2	Create post as standard user(must be logged in)	1.Log in as client with provided test data 2.Navigate to "Appointment" page 3.Enter test data in fields 4.Press confirm button	***LogIn data:*** *Email: "testmail@gmail.com"* *Password:"12345"* ***Test data:*** ***Title: "Test Post"*** ***Body: Optional***
US3	Register	1.Navigate to register page 2.Provide your credentials 3.Press register button	***Register Data:*** *EmailL: "testregister@gmail.com"* *Password:"abc123"*
US5	View post as not logged in user	1.Navigate to Front Page 2.Select a post to view	A post on the front Page of ReReddit.