**Individual Project**

**Test Plan**

**Software Engineering**

**beer.io**

**Fontys**

**University of Applied Sciences**

Prepared by:

Kristian Savov

# What am I testing?

The following parts of the application are going to be tested. That would help with delivering an optimal product. They include but are not limited to:

* Backend extraction testing; This type of testing is essential to ensuring proper database to system intractability
* Backend component integration; This type of testing is enabling us to check if the system components interact as designed
* API testing; Also known as endpoint testing, these tests are going to ensure proper connectivity to any client/headless frontend
* Frontend e2e testing; The frontend components will be tested to ensure proper loading and handling of state

# How am I testing?

In this chapter we will investigate the technological stack and tooling that would enable us to execute the previously mentioned testable components, ensuring quality and long-term maintainability.

* JUnit 5 – unit tests in the backend
* AssertJ – comprehensive assertion
* Mockito – mocking services and database
* Cypress – automated headless e2e testing

# Acceptance tests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | Description | Steps | Test data | Result |
| 1 | Register as a user | 1. Enter data in the displayed fields 2. Press register button | Username: “John” Email: “John@gmail.com”  Country: Netherlands,  Password: “John123”  Confirm Password: “John123” | User is successfully registered and is redirected to login page to login with the newly created account |
| 2 | Login as a user | 1. Enter required data in the displayed fields 2. Press login button | Username: “John”  Password: “John123” | User is successfully authenticated and is redirected to the login page |
| 3 | Send a message as a user | 1. Register, if not done already 2. Login, if not done already 3. Write a message to be sent 4. Press send message button | Message: “Hello, World!” | User has successfully sent the message, which can be seen on the page now. If another user is logged in they will also see the same message |