

## Assignment 4: GUI Testing

**Overview:** Most systems are accessed by a user through a graphical user interface, which also needs to be tested. Manual GUI tests are expensive and do not scale well, so we need automatic GUI tests that simulate the behavior of a real user. Cypress<sup>1</sup> is one of several front-end testing frameworks that fulfill this task.

**General notes:** This assignment counts towards Lab 2. Deliver solutions to all exercises as defined in the *deliverables* to pass this assignment. When using material outside of the official course material (lecture, slides, tutorials), state the source and critically reflect on its usage. In case of questions, please contact Zeth Danielsson (zeda21@student.bth.se).

**Work distribution:** At the beginning of each assignment submission, state how the work was distributed among the two team members.

### 1. Graphical User Interface Tests

Assuming that the backend has been sufficiently tested and is, therefore, reliable, we want to build higher-level scenario tests on top of them to ensure that our GUI offers the functionality as defined in the requirements specification to a human user. *Using Cypress, create automatic E2E tests for the use cases associated with requirement 8 of the EduTask specification, i.e., ensure that a user can create, toggle, and delete to-do items of a task.*

**Deliverables:** The submission to this exercise must contain all of the following:

1. A list of test cases for user cases R8UC1, R8UC2, and R8UC3, which you derived by applying the test design technique (as far as appropriate).
2. An implementation of these test cases using Cypress (provide a link to your Cypress test code in your forked repository).
3. A screenshot of the test execution, including a brief report of system failures if detected.

### 2. Declarative vs. imperative UI Testing

The terms declarative and imperative are most commonly used in software engineering when classifying the paradigm of a programming language. Still, those terms also apply to test case implementation on the UI level. *Explain the two concepts and argue which makes the most sense for UI testing.*

**Deliverables:** The submission to this exercise must contain all of the following:

1. An explanation of declarative and imperative UI test case implementation.
2. A discussion of which of the two approaches is most applicable in UI testing.

---

<sup>1</sup>See <https://www.cypress.io/>