Web testing
What is web testing?

Testing web applications

Applications of which the client runs in a web browser
In this lecture on web testing

What to test

How to test it
What to test

- Security testing
- Load testing
- Functionality testing
  - **End-to-end testing**
  - **Unit testing**
- Performance testing (anything non-functional)

- User acceptance testing
- **Automated user interface testing**
- A/B testing
- Exploratory testing
- **Accessibility testing**

Back end (server) ➔ HTTP ➔ Front end (client)
Tips

• Carefully consider the aforementioned types of tests
• Take all these tests into account when doing hour estimations
The remainder of this lecture

JavaScript unit testing and UI component testing
End-to-end testing
Accessibility testing
Introduction to JavaScript unit testing and UI component testing
Technologies used

- React
- Enzyme
- JEST
- WebStorm

... but pick whatever you like best!
Live coding: JavaScript unit tests

Pay attention to the following:

• How does this compare to your Java unit tests?
• How can you design your JavaScript for testability?
Tips

• Use a purely functional coding style
• Separate logic from UI
• Use modules to organise your code and to allow for easy mocking
React components

Example:

```javascript
const TitleComponent = props => {
  return (
    <h1>{props.title}</h1>
  );
};
```
Live coding: React component tests

• Compare this to “normal” unit tests of functions.
Tips

• Create small, independent components
• Use component tests to do early UI testing
• Snapshot tests can save you a lot of work, but use them with care
End-to-end testing
Story time: testing at QDelft
Story time: testing at QDelft

Example of a manual “test script”:

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
<th>OK/NOK</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Open the web application at <a href="http://localhost:3000">http://localhost:3000</a></td>
<td>OK</td>
</tr>
<tr>
<td>2</td>
<td>Click the search box</td>
<td>OK</td>
</tr>
<tr>
<td>3</td>
<td>Type some text in the search box and check whether the results list is</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>updated automatically</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Check whether the search results match the input</td>
<td>NOK</td>
</tr>
</tbody>
</table>
Automating end-to-end tests

Selenium

- Works as a “remote control” for your browsers.
- Can be used for automating any task (not just testing).
- Useful for cross-browser testing (with services like BrowserStack).

Cypress

- Integrates with your browser and is specifically targeted at end-to-end testing.
Demo: Cypress
Challenges when doing end-to-end testing

• Test data
• Finding elements on the page
• Waiting for results
Accessibility testing
Accessibility

Make sure that anyone can use your web application, including people with disabilities.
Ishihara test
Example: “your indicated availability”

<table>
<thead>
<tr>
<th>Time</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:00-11:00</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>11:00-13:00</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>13:00-15:00</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>15:00-17:00</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

When am I **not** available?
Example: “your indicated availability”

<table>
<thead>
<tr>
<th></th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:00-11:00</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>11:00-13:00</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>13:00-15:00</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>15:00-17:00</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

When am I **not** available?
Example: “your indicated availability”

<table>
<thead>
<tr>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

Success Criterion 1.4.1 Use of Color
Color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.

— Web Content Accessibility Guidelines (WCAG) 2.1
Web Content Accessibility Guidelines (WCAG)

Content should be:
• Perceivable
• Operable
• Understandable
• Robust

https://www.w3.org/TR/2018/REC-WCAG21-20180605/
Accessibility: start early

If you only start thinking about accessibility during the testing phase, you are far too late.

Start with *inclusive design*.

Writing proper HTML prevents many accessibility issues.
Accessibility tools

aXe  WAVE  Tenon
Manual testing

Essential for accessibility testing (tools miss >30% of issues).

- https://accessibility.blog.gov.uk/2017/02/24/what-we-found-when-we-tested-tools-on-the-worlds-least-accessible-webpage/

User interaction, logical flow of the application.
Thanks!

• All the code is available at https://github.com/fm2003/emoji-search
Web testing