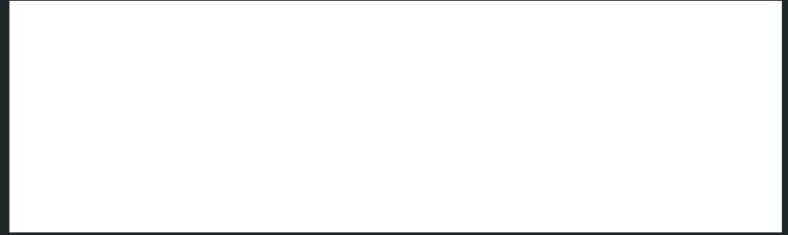# TDTP - Morning Coffee

Contents

1. Information objectives
2. Why we chose particular test design techniques
3. Most important achievements and findings
4. Details about found issues
5. Project conclusions, challenges and thoughts

Information objectives

Part I: The project manager wants you to check for critical bugs that may block the planned release.

Part II: A month after release users report application severe issues that threaten product credibility. The project manager suspects already reported bugs might be the cause and asks your team to investigate them and asks you to design a complex test that may reproduce the failure.

Part III: The project manager wants you to check and test special cases for the delivered features.

Why we chose some particular test design techniques

We have chosen to load test the application to make sure it will run seamlessly, when used by multiple users.

The application may, depending on the limits of its resources, perform completely differently for one user (functional testing) compared to many (load testing).

We have chosen to use scenarios testing because it represents an easier way to test and evaluate end to end complicated problems.

Most important achievements and findings

1. *No reply attack protection. The api does not include a nonce or similar string to prevent a scenario where an attacker runs same request many times.*

2. *Cross-origin resource sharing (CORS) - Arbitrary Origin Trusted The application implements an HTML5 cross-origin resource sharing (CORS) policy that allows access from any domain. Allowing access from arbitrary domains means that those domains can perform two-way interaction with the application via the request. Rather than using a wildcard or programmatically verifying supplied origins, use a whitelist of trusted domains.*

Details about the found issues

- The ones previously mentioned...

  +

- When adding a building, save and cancel button move outside their space;
- Possibly invalid input still adds a building (negative values for safe distance, only whitespace name, etc.) - same for editing
- Some weird behaviour after a lot of login attempts (does not let you log in)
- Error when deleting a building
- Reports show an error message right away
- No messages shown to the user for invalid sign in.

Project conclusions, challenges and thoughts

We saw that testing may be implemented in every stage of software development and that is not just a single phase of software lifecycle. As with the other activities in the software lifecycle, testing has its own unique challenges.

As we started to test the LightHouse application, we faced a lot of challenges, one of the most common ones was the beginning phase when we for the first time implemented Selenium for automated tests using Mozilla browser and it crashed at the loading phase . As we dived deeper and deeper into testing using teamwork and coverage-based, risk-based, activity-based and even evaluation based techniques it was easier for us to overcome the challenges we encountered.

Software testing is an important part of the development process with the ultimate goal to help software designers and developers. For us this real-based project was a great experience and we think it might help us a lot in the future.