

Testing

At their core, tests make sure your application is doing what you intend it to do. They are an automated script to execute your code and check that it did what you expected.

End-to-End Tests simulate user behavior – testing from the user’s perspective. In a web application, they will start the server, fire up a browser, click around, and assert that certain things happening in the browser give us confidence our feature is working.

Cypress End-To-End Testing-

https://www.youtube.com/watch?v=7N63cMKosIE&ab_channel=Fireship

Unit Tests exercise units of code according to their public API. These tests involve creating an instance of a class and calling methods on it with specific inputs. You assert that the methods you called had the desired effect (typically that they returned expected outputs). These tests are fast, stable, and are not tightly coupled to many other parts of the system. They do not, however, give you confidence the overall system is working—just that the unit of code under test is working.

Java Unit Testing with JUnit - Tutorial - How to Create And Use Unit Tests -

https://www.youtube.com/watch?v=vZm0IHciFsQ&ab_channel=CodingwithJohn

Acceptance Tests involve testing that the app meets the requirements and expectations of the stakeholders.

Load Tests involve testing how the app handles heavy traffic and high loads, to identify the bottlenecks and areas of the app that may be slowing down the user experience.

Security Tests involve testing the app for vulnerabilities such as SQL injection, cross-site scripting (XSS) attacks, and other security threats.

Simple Penetration Testing Tutorial for Beginners! -

https://www.youtube.com/watch?v=B7tTQ272OHE&ab_channel=LoiLiangYang

Usability Tests involve testing the app from the user’s perspective to ensure that it is user-friendly and provides a positive user experience.

Accessibility Tests involve testing the app to ensure that it is accessible to users with disabilities and meets accessibility standards.

How I do an accessibility check -- A11ycasts #11 -

https://www.youtube.com/watch?v=cOmehxAU_4s&ab_channel=GoogleChromeDevelopers

Your software is in service to some user, so it’s that user who should drive your work. It is not recommended to use tests to design a user experience, so figure out how the user will use the software before writing tests (either by experimental coding or working with a designer).

- Meghna Jagadeesh added #37 as child task 2 weeks ago
- Meghna Jagadeesh added #38 as child task 2 weeks ago
- Meghna Jagadeesh added #42 as child task 2 weeks ago
- Meghna Jagadeesh changed title from Emergency response guide page to Emergency response guide 2 weeks ago
- Aadil Ahmad changed the description 2 weeks ago
- Meghna Jagadeesh @mxj172 · 35 minutes ago
 - Author Maintainer
 - Be able to search for the relevant emergency you want to find out more info about (e.g. plan of action, advice & tips) and it will redirect you to the relevant section of the page. Possible auto-fill function for the search bar when user types.
- Meghna Jagadeesh changed time estimate to 2d 35 minutes ago
- Meghna Jagadeesh changed time estimate to 1d 33 minutes ago
- Meghna Jagadeesh changed the description just now
- Meghna Jagadeesh added #39 as child task 2 weeks ago
- Meghna Jagadeesh added #40 as child task 2 weeks ago
- Meghna Jagadeesh added #41 as child task 2 weeks ago
- Aadil Ahmad changed the description 2 weeks ago
- Meghna Jagadeesh changed time estimate to 6h 35 minutes ago
- Meghna Jagadeesh changed time estimate to 1d 34 minutes ago
- Meghna Jagadeesh changed the description 1 minute ago

added emergency guide page entity

parent 3c3b5da6

main

No related merge requests found

Pipeline #4580 running with stages

Changes 1 Pipelines 1

Showing 1 changed file with 3 additions and 0 deletions

Hide whitespace changes

Inline

Side-by-side

communityjdl.jdl

+3 -0

View file @b01e7ad4

... 25 26 27 28 29 30 31 32 33 ...

@@ -25,6 +25,9 @@ entity ChatMessage {
content String required
sentDate LocalDate required
}
+ entity EmergencyGuidePage {
+ emergencyType String required
+ }
relationship OneToMany {
ChatRoom{messages} to ChatMessage{room}
}