## Team details

|  |  |
| --- | --- |
| Batch & Team name | MS Batch 1 - Code Warriors |
| Team members | Anirudh Satyam, Gulshan Khara |
| Date | 24/09/2019 |
| Status | In-progress |

Notes:

* In this form, ‘your code’ is the code you own now – i.e., Phase-2 code
* If you link the evidence (e.g., to a file in GitHub), you don’t need to update the doc every time

Modifications that we have done to the code received at the start of Phase 2:

* We have linked the registration/discharge process with the alerting system
* We have modified the logic of the alerting system – giving high/low values
* We give alerts for each specific parameter instead of a general warning for any parameter.
* We have added the following endpoints in the Services :
  + Return list of all patients
  + Return particular patient with respect to ID
  + Modification of delete service

In the UI, we have implemented all the user stories defined to us.

Testing:

* Unit tests written for all the code
* Integration testing has been done, for the backend service and the frontend UI
* UI testing done by using “Coded UI Testing” software in Visual Studio. We have recorded the steps to be carried out in the UI, and have automated this test to give the user an idea of how UI is supposed to work

## Compiler warnings and Linting

|  |  |  |
| --- | --- | --- |
|  | Assertion | Exceptions |
|  | Metric: The warning + linting count is zero |  |
|  | Sustain: The ‘treat warnings as errors’ setting is enabled |  |

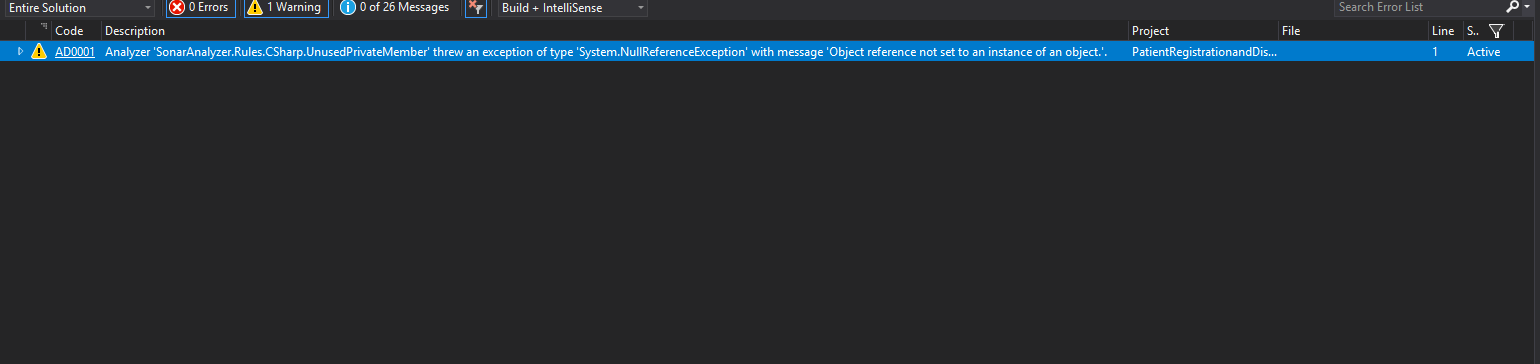
#### What linting tool is used?

Sonar Lint is used

#### How have the warnings fared, since you took over in Phase-2?

All backend warnings were already handled by the previous team. There were a few warnings relating to exceptions, which we have handled.

### Evidence



## Static Analysis / Coding standards

#### What is the max cyclomatic complexity of a function in your code?

3

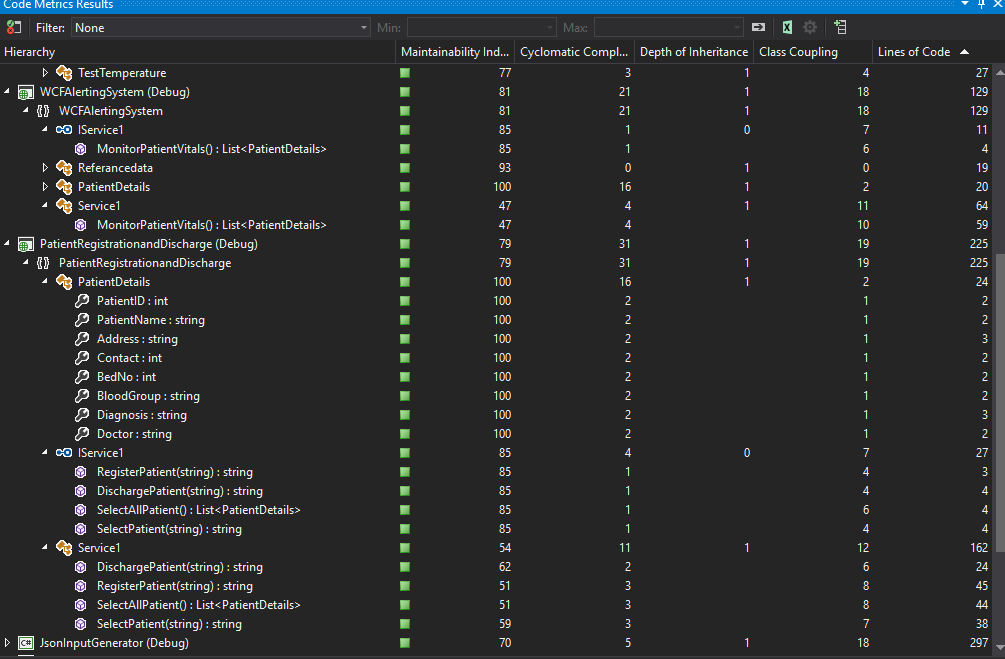
#### Where did you reduce the complexity of the code in Phase2?

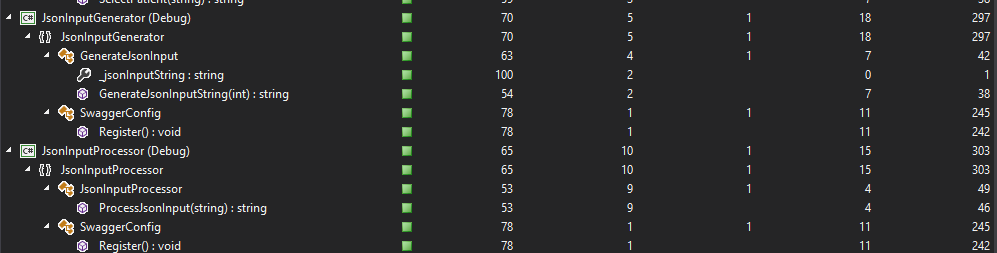
For validating patients, we changed the design and reduced the complexity of the code. It was previously 5.

#### What are the instances where you improved the naming for enhanced clarity / consistency?

Yes, we have changed the names for alert indicator, to enhance readability. Previously the value of alert of each parameter(parameters are pulse,spo2,temperature) were getting overwritten while checking for a critical condition, we have separated them and given them well defined names

### Evidence



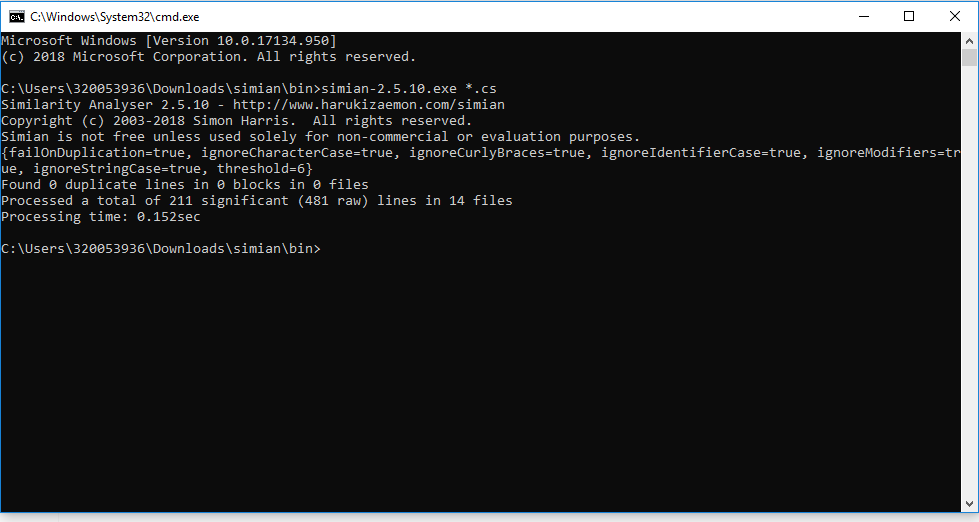


## Duplication

|  |  |  |
| --- | --- | --- |
|  | Assertion | Exceptions |
|  | Metric: The duplication count is zero |  |
|  | Sustain: We have a mechanism to eliminate duplication |  |

#### How much duplication did you reduce in Phase2? List the instances.

### Evidence

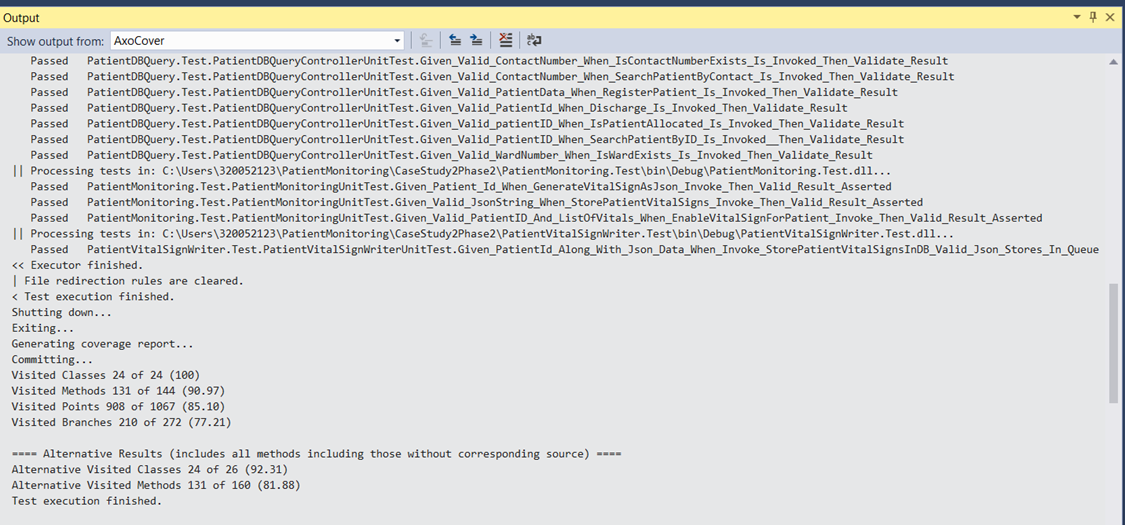


## Unit testing

|  |  |  |
| --- | --- | --- |
|  | Assertion | Exceptions |
|  | Metric: We met our code-coverage target  (if you didn’t fix a target, take 75% as reference) |  |
|  | Sustain: We have a mechanism to enhance coverage and not let it slip. |  |

#### For each not-covered-code: What is the consequence of not-covering?

Explain in terms of development-time, for someone who receives your code



### Evidence

Attach/link: coverage report / screenshot

## Dynamic Analysis

|  |  |  |
| --- | --- | --- |
|  | Assertion | Exceptions |
|  | Metric: We have timing- and usability-related tests |  |
|  | Sustain: We have a mechanism to avoid slippage |  |
|  |  |  |

### Evidence

We have done testing in all levels are per reports already attached in this document.

## Functional / end-to-end tests

#### How many end-to-end tests do you execute, to verify all the functionality?

We have 3 end to end tests to verify the functionality.

We have one test each for

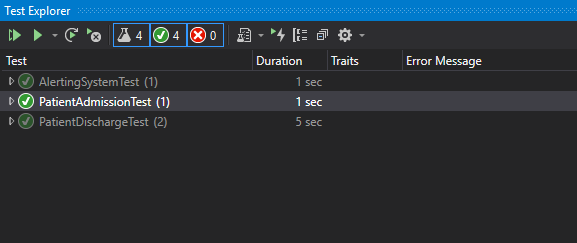
* Patient Registration/Admission
* Patient Alerting System
* Patient Discharge

#### How many are automated?

All of them are automated. We have used “Coded UI Testing” feature which is available in Visual Studio to perform the end to end testing.

### Evidence

Attach/link: test-scripts and/or test-spec



## Retrospection

If you had to do Phase-1 again (with the same time & scope), what would you do different?

* We would have a design before writing the first line of code.
* We would verify the design against the SOLID principles and ensure maintainability and extension of the code becomes easy.
* We would modularize the code, by writing it in separate projects.
* We would have a clear idea about the interactions between various classes and have a sequence diagram to further solidify the interactions.