

## ENSE375-Group D:

Shared Github Repo: <https://github.com/NolanFlegel/ENSE375-GroupD>

### Jenkins pipeline setup:

Jenkins was installed on a pc using docker, with blue ocean running in a separate container. The dashboard is accessible locally through a web browser. A pipeline was created for our shared repository and it scans all branches in the repository for commits. When a commit is detected, the pipeline will execute.

To initiate a Jenkins pipeline, a Jenkins file is created in the root directory of the repository containing all of the settings, stages and steps required by Jenkins. The configuration of the pipeline uses a Maven agent in another docker container that is created temporarily by the Jenkins pipeline. The first stage of the pipeline changes the directory from our repository root to our java project root. A maven pom.xml file was created and saved in the java project root. The second stage of the pipeline runs the command “mvn clean test” where maven will read the pom.xml file, compile the project and execute the junit tests. A report is created and the test results are displayed. This process is designed to be automatic with new builds triggered any time a commit is made in the project repository.

STATUS	RUN	COMMIT	BRANCH	MESSAGE	DURATION	COMPLETED
✓	13	4216dac	main	removed test causing null pointer <span>2 commits</span>	15s	17 minutes ago ↻
✓	1	e1cc755	nolanFixNullptr	Branch indexing	16s	17 minutes ago ↻
✗	1	8d58bc2	nolanFixJUnit4tests	Branch indexing	14s	40 minutes ago ↻
✗	12	1103d16	main	Krupals commit on riskMap <span>2 commits</span>	15s	2 hours ago ↻
✗	2	d7610eb	krupalsBranch	Krupals commit on riskMap	26s	2 hours ago ↻
✓	1	08c0477	nolanFixingPatientTests	Branch indexing	18s	2 hours ago ↻
✓	1	1c6c08b	krupalsBranch	Branch indexing	15s	4 hours ago ↻
✗	11	91cef57	main	Updated patientHistogram bug	13s	4 hours ago ↻
✓	10	1c6c08b	main	fixed PatientHistogram glitch.	13s	10 hours ago ↻
✗	9	be8e70b	main	deleted target/ folder and merging w... <span>4 commits</span>	13s	10 hours ago ↻
✗	3	e46b824	shanes_tests	deleted target/ folder and merging w... <span>8 commits</span>	13s	10 hours ago ↻
✓	8	83c4da0	main	updated PostalCodeTest to use correct version o...	13s	10 hours ago ↻
✗	7	f74affc	main	Junit tests for patient class	13s	11 hours ago ↻
✗	2	b79223d	nolan-riskmeter	Update PostalCode.java & PostalCod... <span>3 commits</span>	13s	11 hours ago ↻
✗	6	21eca7b	main	Update PostalCode.java & PostalCodeTest.java	24s	11 hours ago ↻
✓	1	34b940b	nolan-riskmeter	Branch indexing	18s	11 hours ago ↻
✗	1	cf7e4ba	RenzBranch	Branch indexing	14s	11 hours ago ↻
✗	5	4c144f9	main	Update PatientHistogram.java	11s	a day ago ↻
✗	4	d06b8cc	main	patientHistogram bug fix	11s	a day ago ↻
✗	3	abe6299	main	implemented patientHistogram	13s	a day ago ↻
✓	2	89f4081	shanes_tests	fixed junit configuration <span>2 commits</span>	13s	a day ago ↻
✓	2	f0e7822	main	deleted logs	14s	a day ago ↻
✗	1	3e9ef29	shanes_tests	Branch indexing	13s	a day ago ↻

# Jenkins

Dashboard

New Item

People

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Open Blue Ocean

Lockable Resources

New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

All

S	W	Name	Last Success
		ENSE375-GroupD	6 min 19 sec - <a href="#">log</a>

Icon: [S](#) [M](#) [L](#)

Pipeline Settings

Agent

docker

Image\*

maven:3-alpine

Args

-v /root/.m2:/root/.m2

Environment

Name	Value

Change Directory

Steps

Change current directory

RiskMeter

Shell Script

mvn clean test

+ Add step

Shane Toma

Responsible for:

- PatientList.java
- PatienListTest.java

Summary:

Using TDD I created 8 tests to check for the 8 main input combinations that may cause failure related to my code. While more extensive testing and more thorough error checking may have been prudent, these 8 tests should suffice for our purposes. I tested for

getting, adding, and deleting both null and real patients. I also tested for a number of patients with 0 patients and 3 patients in the list, any other number (other than maybe an extremely large number, this is an additional test I could have planned for) is expected to behave the same as 3. I did not test for incorrectly created patients as this is PatientTest.java's responsibility (this is where adding more tests could have been a prudent redundancy).

#### Jacob Chapman

Responsible for:

- PatientHistogram.java
- PatientHistogramTest.java

Summary:

Using TDD, I implemented 4 tests to account for patient counts being negative and out of bounds in the 2D array (i.e, the available regions). In the source code, There were 2 functions that accounted for adding and deleting patients to a specific region, and 1 for getting a specific count of patients in a region. There were no region details and no patient counts to refer to, so I made the assumption when implementing that there were 100 counts of patients for each region. I did not test for having strings or characters as an index because that was intuitively handled by the syntax.

#### Nolan Flegel

Responsible for:

Patient.java  
PatientTest.java

Summary: I created 13 tests using TDD principles to test the creation and retrieval of patient records. I used regular expression testing to ensure that Patient's names only contain letters of the alphabet and periods I also used regular expressions to ensure that patient ID numbers are exactly 9 digits long and do not contain any zeros. The constructor for the patient class catches any runtime exceptions using try catch statements, throws and lambda functions. I wrote tests to verify that the getter and setter functions are correctly verifying data and performing their functionality. I made an assumption that the acceptable age range is positive integers from 0 to 130. Values outside this range will be rejected and an exception thrown.

#### Renz Rivero

Responsible for:

- PostalCode.java
- PostalCodeTest.java

Summary:

With TDD principles, I implemented 17 tests to ensure that a certain format for a postal code input is followed. This is done by checking the overall input length of the postal

code and checking each index element of it to see if it matches with its expected type. A better implementation would have been to use regular expressions to validate the overall input but for the purposes of this project, this solution will suffice. Given the format "K1a-bxy" to follow, I made an assumption that the dividing character (-) can also be a whitespace character.

Krupalkumar Patel

Responsible for:

RiskCodeMap.java

RiskCodeMapTest.java

Summary:

I had used TDD principles to implement the test code for the riskmap. The file RiskCodeMap contains three main functions in which the definition of the RiskMap is made using a constructor with a 2 dimensional character array. Second function is implemented to find the mean of the list of the neighbour case count and compare it. It then converts the count to the colour associated with it. The test code was implemented by making an object of the RiskCodeMap class and then an object of the Integer array. Adding integers to the array and then updating the function. After this the function is compared and checked which number it contains so that a respective colour can be associated with it. I had to make assumptions on the count of the number of cases. 0 - Green, (0:3] Blue, (3,:7] Yellow, (7,10] Orange, (10:inf] Red.