

1 Team Name:

Jets

2 Team Profile:

The team consists of 4 individuals named Julian Falzon, Ethan Lethridge, Andre Theuma and Luke Dalli.

Julian's technical skill involves that of Management. These include technical writing, computer literacy in database systems and the adobe suite. He has also gained experience in data management, analytics and project management. Julian has leadership skills which are considered as an asset for his position.

Ethan's area of expertise is that of Business Analysis. Some of the skills gained from experience include being fluent in programming languages such as Java, C, C++ and Python. Besides that he is known for his success in data analytics and data modelling. Ethan's consistency backs up his observations. His logical reasoning makes everything seem easy.

Andre is known for his reputation in Project Management. He is excellent at technical reporting and work scheduling software. Works well with Ethan in data analytics and proficient in Scrum and Agile. Other soft skills include, good time management, adaptability and teamwork.

Luke is very efficient in Information Technology. Some of his specific skillsets include, full stack development, cloud computing, network structure and security. Luke is a problem solver. His critical thinking and creative mindset helps to complete the package.

2.1 GitHub Repository

URL: <https://github.com/andreTheuma/jenkinsJets>

2.2 Jenkins Project

"Project Jets" URL: <https://jenkins-ict.research.um.edu.mt/job/Jets/>

3 Source Code Listing

3.1 CmakeLists.txt

```
cmake_minimum_required(VERSION 3.15)
project(calculatorLibrary C)
set(CMAKE_C_STANDARD 11)
add_library( multiply.h division.h add.h Subtract.h)
set(file1 library.c )
add_executable(CalLibrary $file1)
```

3.2 library.c

```
#include "multiply.h"
#include "Subtract.h"
#include "add.h"
#include "division.h"
#include <stdio.h>
int main(void) {
    int option;
    int status;
    float num1, num2;

    do {
        printf("1.Addition2.Multiplication3.Subtraction4.Division");
        printf("\nPlease enter an option");
        status = scanf("%d; &option);
        switch(option) {
            case 1:
                printf("\nAdding");
                printf("\nPlease enter two numbers to add together:");
                scanf("%f; &num1);
                scanf("%f; &num2);
                printf("\nThe addition of %.2f and %.2f is %.2f",num1,num2,add(num1,num2));
                break;
            case 2:
                printf("\nMultiplying");
                printf("\nPlease enter two numbers to multiply by each other");
                scanf("%f; &num1);
                scanf("%f; &num2);
                printf("\nThe multiplication of %.2f and %.2f is %.2f",num1,num2,multiply(num1,num2));
                break;
            case 3:
                printf("\nSubtracting");
                printf("\nPlease enter 2 numbers to subtract the second from the first:");
                scanf("%f; &num1);
                scanf("%f; &num2);
                printf("\nThe result of %.2f minus %.2f is %.2f",num1,num2,Subtract(num1,num2));
                break;
            case 4:
                printf("\nDividing");
                printf("\nPlease enter 2 numbers to divide the second from the first:");
                scanf("%f; &num1);
                scanf("%f; &num2);
                printf("\nThe result of %.2f divided by %.2f is %.2f",num1,num2,division(num1,num2));
```

```
break;

default:
break;
}
} while (status == 1);
return 0;
}
```

3.3 add.h

```
float add(add1, add2) {
return add1+add2;
}
```

3.4 subtract.h

```
float Subtract(float x, float y)
{
return x - y;
}
```

3.5 multiply.h

```
float multiply(float mult1, float mult2)
{ return mult1 * mult2;
}
```

3.6 division.h

```
float division(float div1, float div2)
{
return (div1/div2);
}
```

3.7 Screenshots of Jenkins Setup

Source Code Management

☐ None
☐ CVS
☐ CVS Projectset
☒ Git

Repositories

Repository URL:

Credentials: [Add](#)

[Advanced...](#)

[Add Repository](#)

Branches to build

Branch Specifier (blank for 'any'):

[Add Branch](#)

Repository browser:

Additional Behaviours: [Add](#)

☐ Mercurial
☐ Subversion

Figure 1: Jenkins Setup

Jenkins Config [Jenkins] | Jenkins-ict.research.um.edu.my

General | **Source Code Management** | Build Triggers | Build Environment | Build | Post-build Actions

☐ Restrict where this project can be run [Advanced...](#)

Source Code Management

☐ None
☐ CVS
☐ CVS Projectset
☒ Git

Repositories

Repository URL:

Credentials: [Add](#)

Name:

Refspec:

[Add Repository](#)

Branches to build

Branch Specifier (blank for 'any'):

[Add Branch](#)

Repository browser:

Additional Behaviours: [Add](#)

☐ Mercurial
☐ Subversion

Build Triggers

[Save](#) [Apply](#) [Cancel](#) [Help](#)

Figure 2: Jenkins Setup

Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts)
- ☐ Build after other projects are built
- ☐ Build periodically
- ☒ Poll SCM

Schedule */1 * * * *

⚠ Do you really mean "every minute" when you say "*/1 * * * *"? Perhaps you meant "H * * * *" to poll once per hour

Would last have run at Friday, 29 November 2019 11:36:26 o'clock CET; would next run at Friday, 29 November 2019 11:36:26 o'clock CET.

Ignore post-commit hooks ☐

Figure 3: Jenkins Setup

Post-build Actions

E-mail Notification
X
?

Recipients luke.scerri.19@um.edu.mt julian.debattista.15@um.edu.mt

Whitespace-separated list of recipient addresses. May reference build parameters like \$PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable.

☒ Send e-mail for every unstable build
☐ Send separate e-mails to individuals who broke the build

Add post-build action ▼

Figure 4: Jenkins Setup

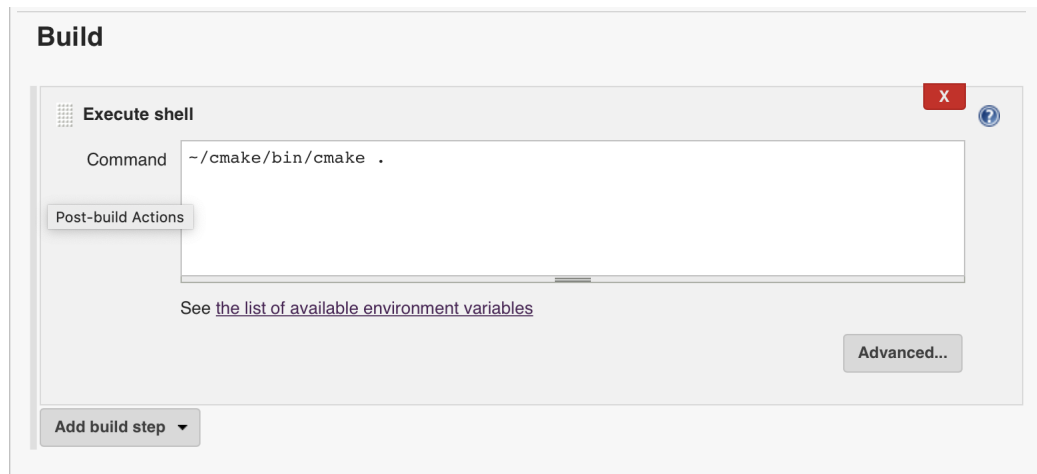


Figure 5: Jenkins Setup