./

Learning Report – Applied SDLC and Software Testing

Course Code: <CODE>



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved By** | **Remarks/Revision Details** |
| 1.0 | 17/09/2020 | Deeksha P | Harsheet Dubey and Sri Rishi Bharadwaj | Pagala Prithvi Sekhar and Srinivas K | The system development life cycle was developed for our model |
| 1.1 | 18/09/2020 | Deeksha P | Harsheet Dubey and Sri Rishi Bharadwaj | Pagala Prithvi Sekhar and Srinivas K | Requirements was amended and agile model was developed for same |
| 1.2 | 19/09/2020 | Deeksha P | Harsheet Dubey and Sri Rishi Bharadwaj | Pagala Prithvi Sekhar and Srinivas K | CI for C program was designed and developed |

**Document History**

# 

Table of Contents

[Activity 1 - System/Software Development 5](#_Toc52141894)

[1.1 REQUIREMENTS 5](#_Toc52141895)

[1.1.1 Aging and Cost gradation 5](#_Toc52141896)

[1.1.2 Problem Statement 7](#_Toc52141897)

[1.1.3 SWOT Analysis 8](#_Toc52141898)

[1.1.4 Requirements 8](#_Toc52141899)

[1.2 Design 9](#_Toc52141900)

[1.2.1 High Level design 9](#_Toc52141901)

[1.2.2 Low level design 12](#_Toc52141902)

[1.3 TEST PLAN 14](#_Toc52141903)

[Activity 2 – Agile Aspects 16](#_Toc52141904)

[2.1 Theme 16](#_Toc52141905)

[2.2 Epic 16](#_Toc52141906)

[i. RTL Description – The Verilog code is developed for the application 16](#_Toc52141907)

[ii. Synthesis – netlist generated is verified and used as input for Physical Design 16](#_Toc52141908)

[iii. Physical Design – Layout for the application is generated 16](#_Toc52141909)

[iv. Algorithm – a hybrid algorithm is chosen and amended on the hardware 16](#_Toc52141910)

[v. Signoff Checks – The layout generated should undergo these signoff checks 16](#_Toc52141911)

[2.3 User Story 16](#_Toc52141912)

[Activity 3 - CI for Applied SDLC and Testing 18](#_Toc52141913)

[3.1 Requirements 18](#_Toc52141914)

[3.1.1 Aging and Cost Gradation 18](#_Toc52141915)

[3.1.2 Problem Statement 19](#_Toc52141916)

[3.1.3 Requirements 19](#_Toc52141917)

[3.2 Design 20](#_Toc52141918)

[3.2.1 High Level Design 20](#_Toc52141919)

[3.2.2 Low Level Design 21](#_Toc52141920)

[3.3 Test Plan 22](#_Toc52141921)

[3.4 CI Workflow for SMART Band C program 23](#_Toc52141922)

[3.4.1 GIT 23](#_Toc52141923)

[3.4.2 Make and Build 24](#_Toc52141924)

[3.4.3 Code Quality 25](#_Toc52141925)

[APPENDIX 26](#_Toc52141926)

[1. C program on GIT 26](#_Toc52141927)

[2. CI for C program of STEPin module 26](#_Toc52141928)

[2.1 GIT 26](#_Toc52141929)

[2.2 Make and Build 28](#_Toc52141930)

[2.3 Code Quality 29](#_Toc52141931)

[REFERENCES 29](#_Toc52141932)

**List of Figures**

[Figure 1: Classification of Cryptography 5](#_Toc52142133)

[Figure 2: Activity Diagram 9](#_Toc52142134)

[Figure 3: Communication Diagram 9](#_Toc52142135)

[Figure 4: Class Diagram 10](#_Toc52142136)

[Figure 5: Object Diagram 10](#_Toc52142137)

[Figure 6: Communication Diagram 11](#_Toc52142138)

[Figure 7: Activity Diagram 11](#_Toc52142139)

[Figure 8: Object Diagram 12](#_Toc52142140)

[Figure 9: Class Diagram 12](#_Toc52142141)

[Figure 10: Activity Diagram 19](#_Toc52142142)

[Figure 11: Communication Diagram 20](#_Toc52142143)

[Figure 12: Class Diagram 20](#_Toc52142144)

[Figure 13: GIT repository and badges 22](#_Toc52142145)

[Figure 14: Commits made in GIT repo 23](#_Toc52142146)

[Figure 15: Unit testing build 23](#_Toc52142147)

[Figure 16: Make file 24](#_Toc52142148)

[Figure 17: Code quality report 24](#_Toc52142149)

[Figure 18: Commits made 26](#_Toc52142150)

[Figure 19: Folders in GIT repo 26](#_Toc52142151)

[Figure 20: Screenshot of build 27](#_Toc52142152)

[Figure 21: Make file 27](#_Toc52142153)

[Figure 22: Code quality 28](#_Toc52142154)

**List of Tables**

[Table 1: Key analysis of all existing algorithms 7](#_Toc52142027)

[Table 2: SWOT analysis 8](#_Toc52142028)

[Table 3: Requirement mapping for high level and low level requirements 14](#_Toc52142029)

[Table 4: Test plan 15](#_Toc52142030)

[Table 5: High level requirements 19](#_Toc52142031)

[Table 6: Low level requirements 20](#_Toc52142032)

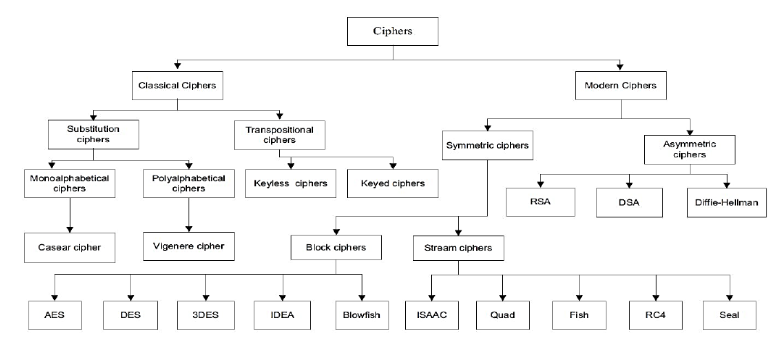
[Table 7: Test plan 23](#_Toc52142033)

**Topic - ASIC DEVELOPMENT FOR IMAGE ENCRYPTION**

# Activity 1 - System/Software Development

## REQUIREMENTS

### 1.1.1 Aging and Cost gradation



**Figure 1: Classification of Cryptography**

1900: Classical

* Substitution-

Monoalphabetical (Casear cipher)

Polyalphabetical (Vigenere cipher)

* Transpositional-

Keyless

Keyed

1980: Modern

* Symmetric-

Block (Blowfish, 3DES, DES, AES)

Stream (RC4)

* Asymmetric-

(RSA, DSA, Diffie-Hellman)

1987: RSA

1987: RC4

1993: Blowfish

1997: DES

2000: AES

**Key Space Analysis:**

|  |  |
| --- | --- |
| **Cipher** | **Key Length** |
| Vigenere | 128 |
| RC4 | 256 |
| DES | 56 |
| 3DES | 168 |
| AES | 128 |
| Chaotic map | 3\*50 |
| Hyper chaotic map | 273 |

Table 1: Key analysis of all existing algorithms

**Cost**

1. **Classical**

* Substitution – low cost but less secure
* Transitional – high cost and time consuming

1. **Modern**

* Symmetric – low cost and high performance
* Asymmetric – low cost less secure

High cost & high logical effort

Low cost with high performance

Low cost with more computational cycles

### 1.1.2 Problem Statement

In the growing information era, information distribution and transferal has amplified exponentially. Images extremely donate to communication during this age of the multimedia. Safety, integrity, privacy, and verification services are the most significant factors in information security. Encryption is a method used to preserve image privacy. This report presents a comprehensive study of various techniques for image encryption. The existing technology is analyzed with respect to algorithm, type of implementation, memory management, and mathematical computations. The main aim of this report is to obtain a hybrid solution for the existing drawbacks and develop an enhanced system for image encryption and decryption. An ASIC development for image encryption.

### 1.1.3 SWOT Analysis

|  |  |
| --- | --- |
| **Strength-**   * Open access tools * Process verification * VLSI based skillset along with RTL coding * Business benefit * Budget cost for FAB | **Weakness-**   * Outsourcing of project required * Brute force attacks |
| **Opportunity-**   * FAB done on business demand * New innovations * No external infrastructure * Semiconductor consumption, foundry are market opportunities | **Threats-**   * Existing time and power constraints * Quantum cryptography * No environment affects * Issue only during fabrication |

Table 2: SWOT analysis

### 

### 1.1.4 Requirements

**High End Requirements**

|  |  |
| --- | --- |
| **ID** | **Description** |
| HL\_01 | The clock frequency & Operating frequency should be 1GHz |
| HL\_02 | Operating voltage used is nominal for the power requirement of 0.8V |
| HL\_03 | The strength of encryption depends on the application then the corresponding key size is chosen to be 128 bits |
| HL\_04 | For developing an ASIC the technology used is 45 nm for better optimization |

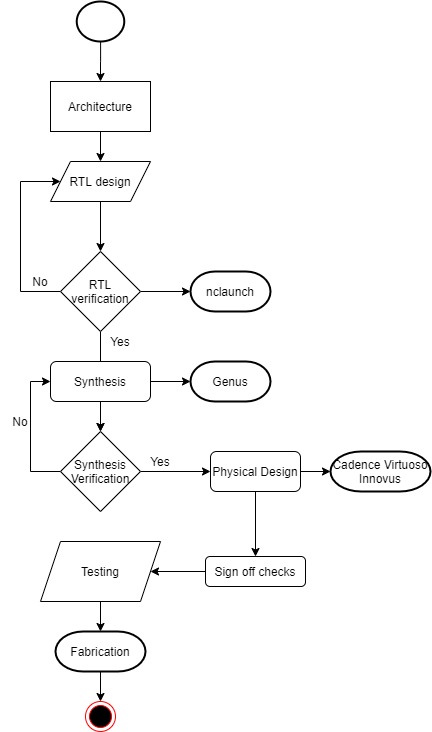
**Low End Requirements**

|  |  |
| --- | --- |
| **ID** | **Description** |
| LL\_01 | The number of cells required to place in standard cells is 19483 |
| LL\_02 | Hardware accelerator required for enhanced performance by developing an ASIC |
| LL\_03 | The throughput i.e pace of input reaching output should be 2.5Gbps |
| LL\_04 | Verilog coding is used for RTL coding |
| LL\_05 | Power requirement for the ASIC development is 1.29nW |
| LL\_06 | Area consumption of the standard cells should not exceed 2,70,400 um\*um |

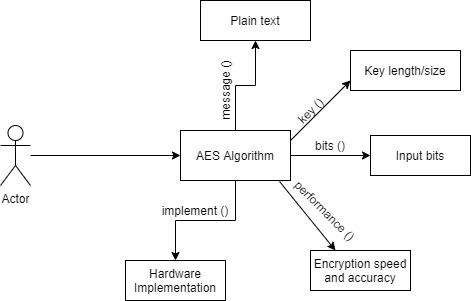
## Design

### 1.2.1 High Level design

**Behavioral Diagram**

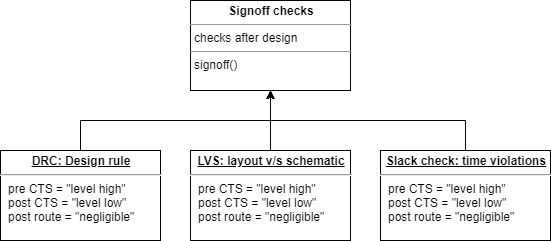


**Figure 2: Activity Diagram**

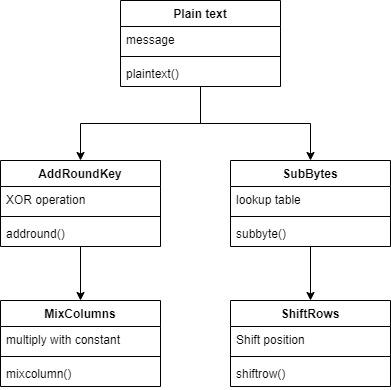


**Figure 3: Communication Diagram**

**Structural Diagram**



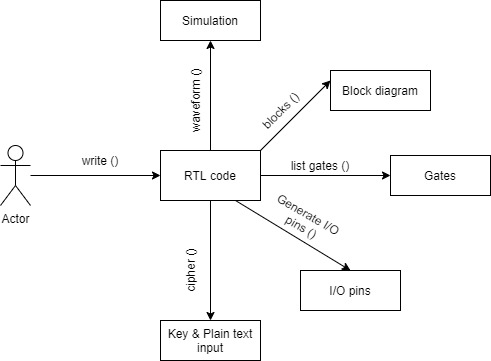
**Figure 4: Class Diagram**



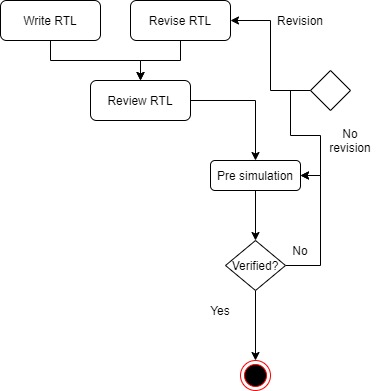
**Figure 5: Object Diagram**

### 1.2.2 Low level design

**Behavioral Diagram**

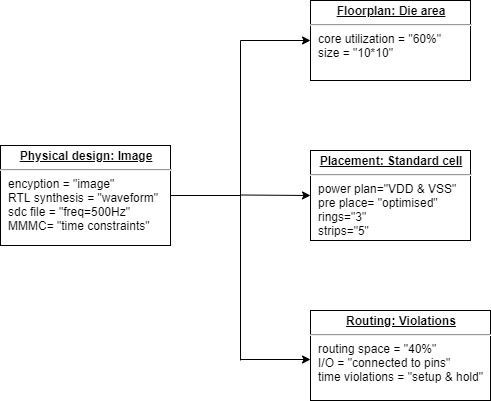


**Figure 6: Communication Diagram**

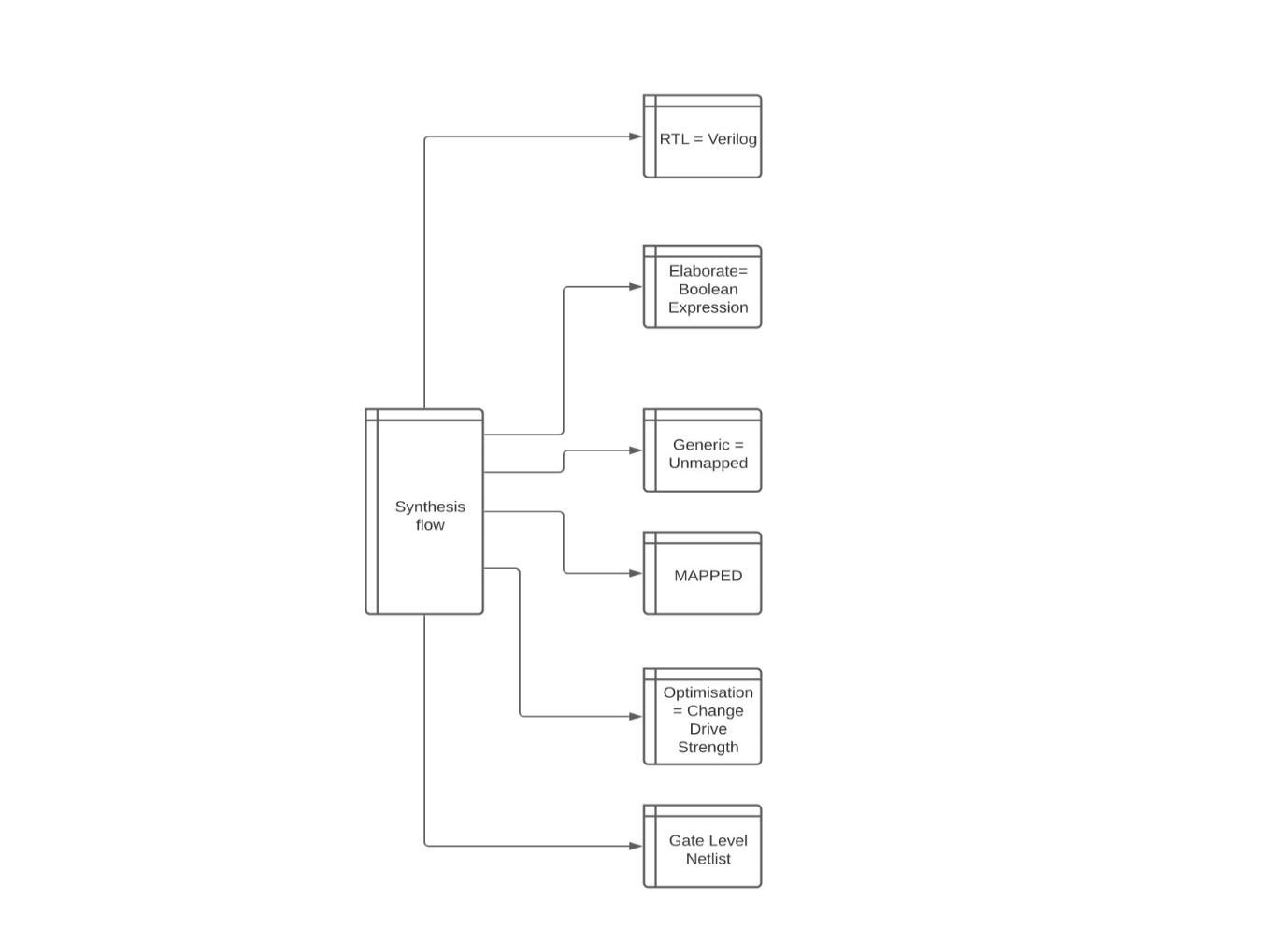


**Figure 7: Activity Diagram**

**Structural Diagram**



**Figure 8: Object Diagram**



**Figure 9: Class Diagram**

## TEST PLAN

**Requirement Mapping**

|  |  |
| --- | --- |
| ID | Description |
| H01\_L03 | The operating frequency of 1GHz is met by keeping the slack in terms of positive or zero which gives the clock period to be set as 1000ps. |
| H02\_L05 | Operating voltage is kept 0.8V which is met by keeping the power constraint limited to 1.29nW |
| H03\_L02\_L04 | The key bits required in development of Hardware accelerator is 128 bits which uses AES algorithm principle. The hardware Description language for RTL is Verilog coding. |
| H04\_L01\_L06 | The physical design development stage uses a IC development technology of 45nm. This meets the silicon width maintenance and accommodates more number of cells of 19483. Area is optimized by using a 45nm technology which is 2,70,400 um\*um |

Table 3: Requirement mapping for high level and low level requirements

**Test plan requirement**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **DESCRIPTION** | **PRE-CONDITION** | **EXPECTED INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** |
| 1. | RTL description | The libraries required for the Verilog coding is set and mapped onto the code | The Verilog code, input pin description, plain text input | Error free compiling RTL with cipher text generation | Cipher text is generated with slight variations due to 128 bit long plain text input |
| 2. | Algorithm used | The input and key size to be mentioned prior to the operation which is dependent on the application | The number of rotations to perform the encryption. A default value is set after deciding the key size length | Encrypt/Decrypt the image successfully | Successful encryption and decryption which can be tested using the external image bit read module |
| 3. | Synthesis | The verified RTL code is free of any errors | The input and output terms mapped to respective hardware accelerator ports | Gate level netlist generation which describes about the gates used for mapping onto the hardware accelerator | Successful generation of netlist but complex gates used was unmapped in gtech which can be cleared using gpdk045 library. |
| 4. | Floorplanning | The libraries for cadence innovus is initially set | The core utilization is specified which is 60% for our project | The core area and die area is created with a micro-meter sq. dimension | 10\*10 grid is developed |
| 5. | Power planning | The initial vdd and vss supply is defined by default | The number of strips and rings are specified | The strips of vdd and vss are generated to provide power supply | The vdd and vss strips might short in some cases |
| 6. | Routing | The standard cell placement and routing area in the layout is specified | The utilization for routing is 40% | The routing stage removes all the drc violations and sets the routing paths for current flow | The routing space depends on the connectivity. It fails only when the space is crowded |

Table 4: Test plan

# Activity 2 – Agile Aspects

## 2.1 Theme

**ASIC Development for Image Encryption**

In the growing information era, information distribution and transferal has amplified exponentially. Images extremely donate to communication during this age of the multimedia. Safety, integrity, privacy, and verification services are the most significant factors in information security. Encryption is a method used to preserve image privacy. The existing technology is analyzed with respect to algorithm, type of implementation, memory management, and mathematical computations.

When either communication bandwidth or storing is restricted, information is often compacted. In specific, when a wireless communication system is working, low-bit-rate compression algorithms are needed as outcomes of bandwidth limits. The major resolution of cryptography is to make definite confidentiality, truthfulness and convenience through the appliance of encryption methods

## 2.2 Epic

Due to the complexity of the project we break down the entire theme into sub-systems. The main goal for developing an ASIC is to focus on 3 major parameters- Area, Power, and Time. The RTL simulation results are discussed and verified in this paper. The flow of ASIC design is proposed from RTL to GDSII.

1. RTL Description – The Verilog code is developed for the application
2. Synthesis – netlist generated is verified and used as input for Physical Design
3. Physical Design – Layout for the application is generated
4. Algorithm – a hybrid algorithm is chosen and amended on the hardware
5. Signoff Checks – The layout generated should undergo these signoff checks

## 2.3 User Story

1. The RTL stage consists of blocks for performing encryption/decryption written in Verilog code. The behavior and design function are defined in Hardware Description Language (HDL) form during this stage. The RTL consists of main block for AES core and four sub-blocks for performing AES operation which consists of s-box implementation, shiftrows, mixcolumns and addround keys. All these sub-blocks are instantiated in the main block. RTL is tested in terms of the input bits and the ports labelling if these cases fail then the RTL is again revised. Additional checking for RTL is performed to test the syntax of RTL and its functionality. The simulated waveform is verified with the online AES calculator. When the design's functionality does not fulfill the requirements, then we must again conduct the functional design to satisfy the design requirements. The RTL will contain all the expectations from customer in the form of a code written using Verilog. The functionality of the final result is depicted through RTL.

The time duration for developing this sub-system is ***2-3 weeks***.

**As an** RTL vendor

**I want** to see a error free running code

**So that** no issues arise in PD stage

1. Logic synthesis, it is a method of translating the verilog codes of RTL into optimized netlist generation at gate-level. Verification of generated netlist is carried out. The subsequent phase, after functional verification phase, comes the logic design phase. In the process of logic synthesis we design the system's logical networks by following the design's functional descriptions. During this step, the RTL is essentially translated to logic-level representation. Moreover, the logic test is used to verify the netlist correctness. From design specification to logic testing, the design process phases belong to the ASIC front end system. logical optimization and static timing analysis (STA) are performed to assure the compliance with timing, power, strength, area and other restraints. When the Genus has been started the compiler must run the script file instructions. The gate level netlist is mapped and verified with RTL and it acts as input to the Physical design.

The time duration for developing this sub-system is ***2.5 weeks***.

**As an** Simulation vendor

**I want** to see a RTL code generating the required reports with gate level netlist. The verification of same should be compared both with waveform and online AES calculator

**So that** timing violations are neglected

1. The Physical Design stage is after the generation of the optimized netlist is physical design. The optimized netlist at gate-level is executed into the layout at this section. The physical design process of the ASIC development inputs the circuit design as an input and produces the circuit layout as their output. It is achieved in many phases under the back-end design section including partitioning, floor-planning, power planning, placement, routing, and compaction. The initial step is to setup the gpdk045 which contains all rules of standard cells and verifies it while giving input in MMMC file. The core utilization and die area and size is specified before starting the design.

The time duration for developing this sub-system is ***4-5 weeks***.

**As an** PD Engineer/ Semiconductor vendor

**I want** to see a physical design developed for the project which meets the requirements and the RTL specifications

**So that** the cost of fabrication is reduced and fault debugging is passed successfully.

**Topic - MULTIPURPOSE SMART BAND**

# Activity 3 - CI for Applied SDLC and Testing

## Requirements

### 3.1.1 Aging and Cost Gradation

**Aging**

* **1526** – Ancient earlier portable watches were invented and manufactured by British govt. accuracy could mean off by more than half an hour per day, and most didn’t even have a minute hand.
* **1720** – First accurate pocket watch. The balance wheel and modern escapement finally made timepieces useful for science and navigation.
* **1910** – pocket watches move to be wrist watches. During World War I soldiers began to modify pocket watches with straps – and soon after the men’s wristwatch were really born.
* **1926** - Wristwatches become water and dust resistant.
* **1960** - Electronic watches were born. The first electronic watches used “humming” tuning forks powered by a battery.
* **1970** - Quartz watches enable cheap, ultra-accurate timekeeping. The landed Swiss watchmaking elite were divided on the electronic revolution’s place in the future of watchmaking – which ended up being a disruptive force that almost toppled the mechanical watch industry.
* **1980** - Computers and clocks Combine: The Calculator Watch was born. The size was huge with inaccuracy.
* **1990** - The connected watch is born, later dies: The SPOT watch. Popular culture has imagined intelligent, connected devices worn on the wrist since the early 20th century, but it was not until the mobile phone era, when connected watches that offered useful information were available. Using radio signals, the original SPOT watches could offer things like news, weather, stock prices and sports scores for a modest subscription price.
* **2010** - The connected watch was reborn. It had the functionality of connecting it to your smart phones. The batter backup was a major issue. It showed a hybrid functionality of messages, playing music, camera, and other screen evolution technologies were supported.

**Cost Gradation**

Cost=500-30000

Cost=300-10000

Cost = 90-500

Cost=150-5000

### 3.1.2 Problem Statement

* To develop an enhanced model of smart band this will perform multipurpose functions.
* The main aim of this model is to have access to perform all mathematical, logical, and arithmetic operations by building a handy portable band.
* It serves multiple ways of carrying. Can be used as a watch, or a miniature pod and can be used to stick on walls, boards or notebooks.
* It also has a feature of a projector for keypad which helps in flexible input entry.

### 3.1.3 Requirements

**High End Requirements**

|  |  |
| --- | --- |
| **ID** | **Description** |
| HL\_01 | Display screen with a portable screen width |
| HL\_02 | Straps for wrist band operation or when operated as a stick device the necessary casing and stand |
| HL\_03 | Pico projector along with proximity sensor for projecting |
| HL\_04 | Wifi module for Connection to cloud to store history of calculations |
| HL\_05 | Memory card + ROM with a micro USB port |

Table 5: High level requirements

**Low End Requirements**

|  |  |
| --- | --- |
| **ID** | **Description** |
| LL\_01 | Processor to perform all mathematical operations and interconnecting with other modules. Embeds the software onto this hardware module |
| LL\_02 | Battery for power supply along with chargeable ports |
| LL\_03 | Vibrator which notifies the band while giving the result or when any misuse function happens it sends an alert signal |
| LL\_04 | The number of operands for performing any calculation is restricted to 3 operators. |
| LL\_05 | Only integer values can be entered for calculating any operation. |
| LL\_06 | The chargeable time for the band is 20mins with a battery backup of 3-4 days |

Table 6: Low level requirements

## 3.2 Design

### 3.2.1 High Level Design

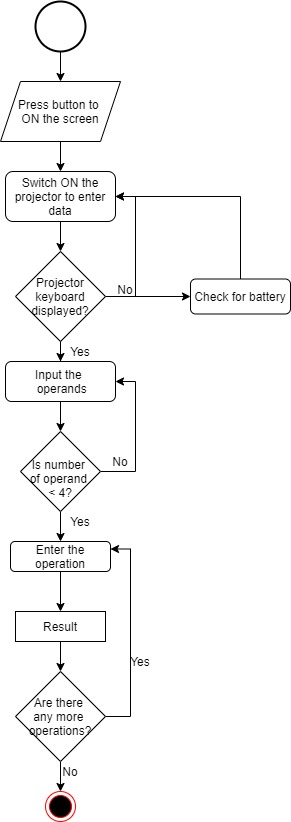


Figure : Activity Diagram

### 3.2.2 Low Level Design

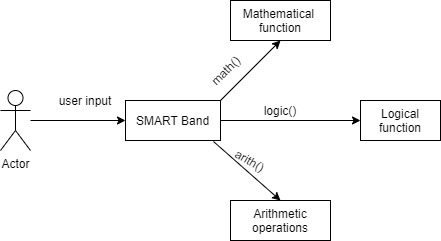


Figure : Communication Diagram

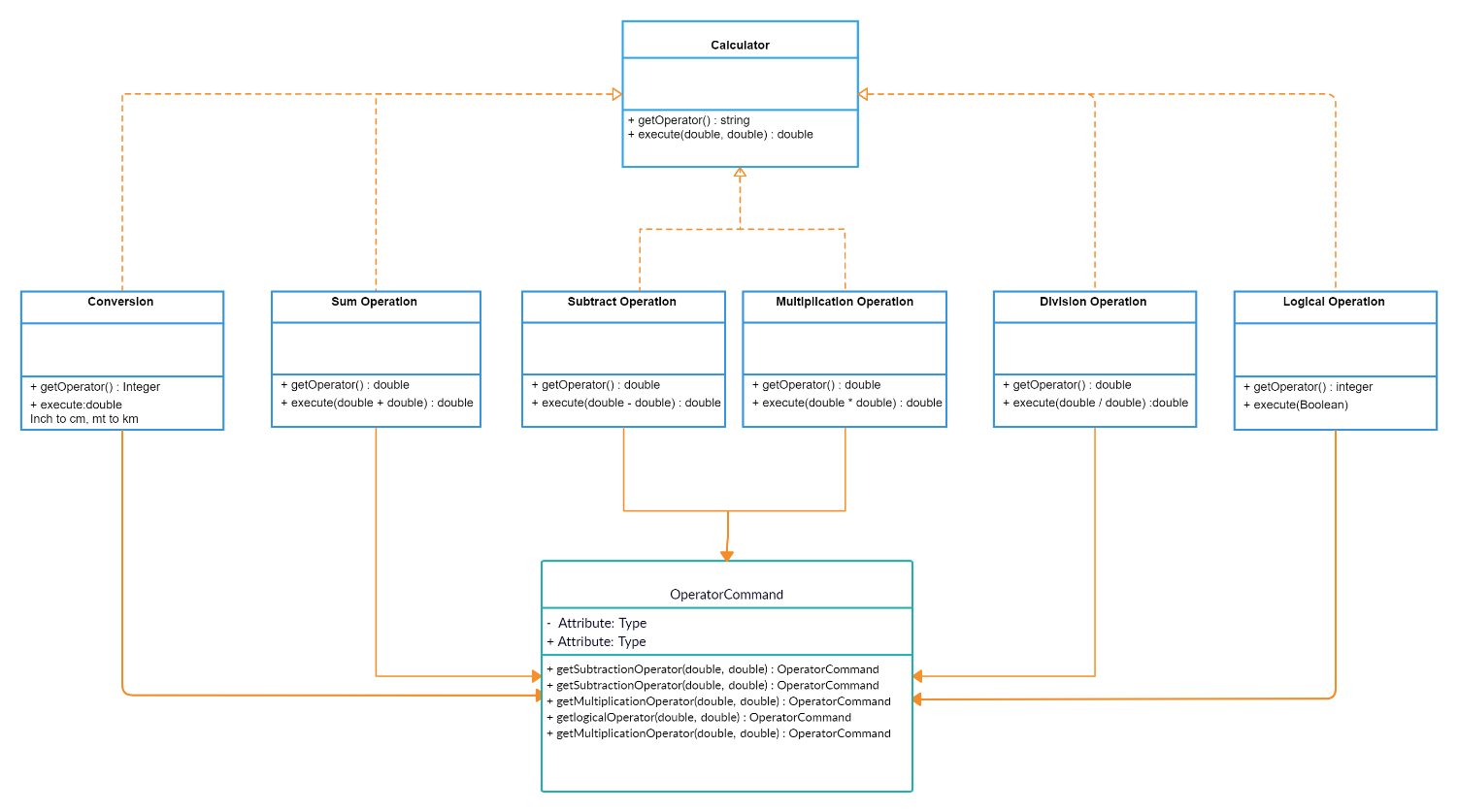


Figure : Class Diagram

## 3.3 Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Pre-Condition** | **Expected input** | **Expected Output** | **Actual Output** |
| 1. | Fetch the initial user input for operands | The max number of inputs can be 3 | Integer values | Ask for the operation to be performed | Ask for the operation to be performed |
| 2. | Fetch user input for operation | The operation number is assigned based on the case statements | Respective operation’s case number | The function to perform that operation is called in main program | The respective operation is carried out on the operand |
| 3. | Operations with one operand | Assign data enter in sequence of the operands specified | When only one operand is inputted instead of three | It should perform single operation on the operand after entering the resp. operation function | Considering single operand as input and carry out the operations |
| 4. | Multifunction on same operands | Data input | When more than one operation to be performed on the operand before exiting | It should perform all the operations and ask for next operation choice before quit | Asks for choice before exiting |
| 5. | Check for any number when divided by zero | Any divide operation | Number divided by zero | Should show error and quit | Will return 0 and exit the loop |
| 6. | Prime number check | A prime function operation | 1 | Should return neither prime not composite | Returns neither prime nor composite |
| 7. | Modulus check | Modulus function with 2 operands | A number modulus zero | Should show error and return exit | Shows error with an exit |
| 8. | Check size | Refer the requirement | The user can give input flexibly | Small size with easy touch screen operation | Handy and easy accessible |
| 9. | Pico projector | The sensors are mounted | Proximity sensors are used to project over surface | The keypad is easily accessible on the projected keyboard |  |
| 10. | Factorial check | Conditions for 0 & 1 factorials | 0 | The factorial of 0 should give output as error | Results in error and exits the loop |

Table 7: Test plan

## 3.4 CI Workflow for SMART Band C program

### 3.4.1 GIT

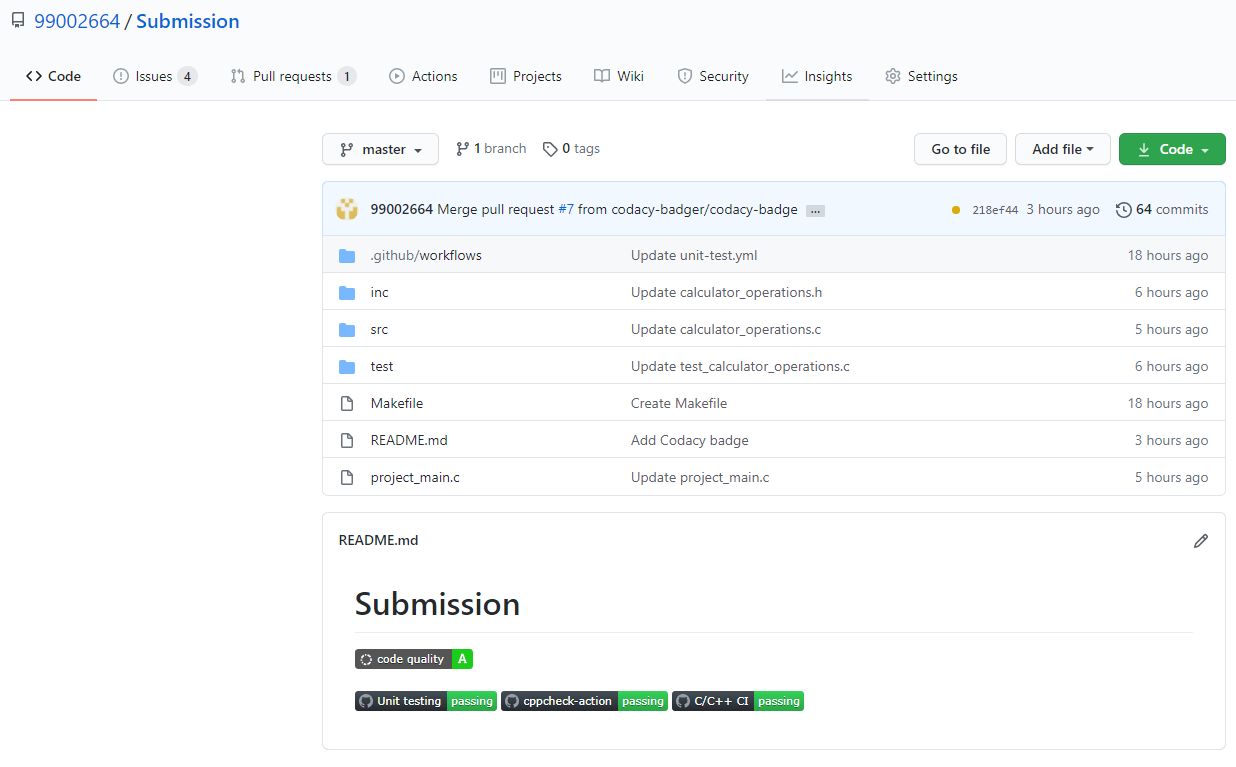


Figure 13: GIT repository and badges

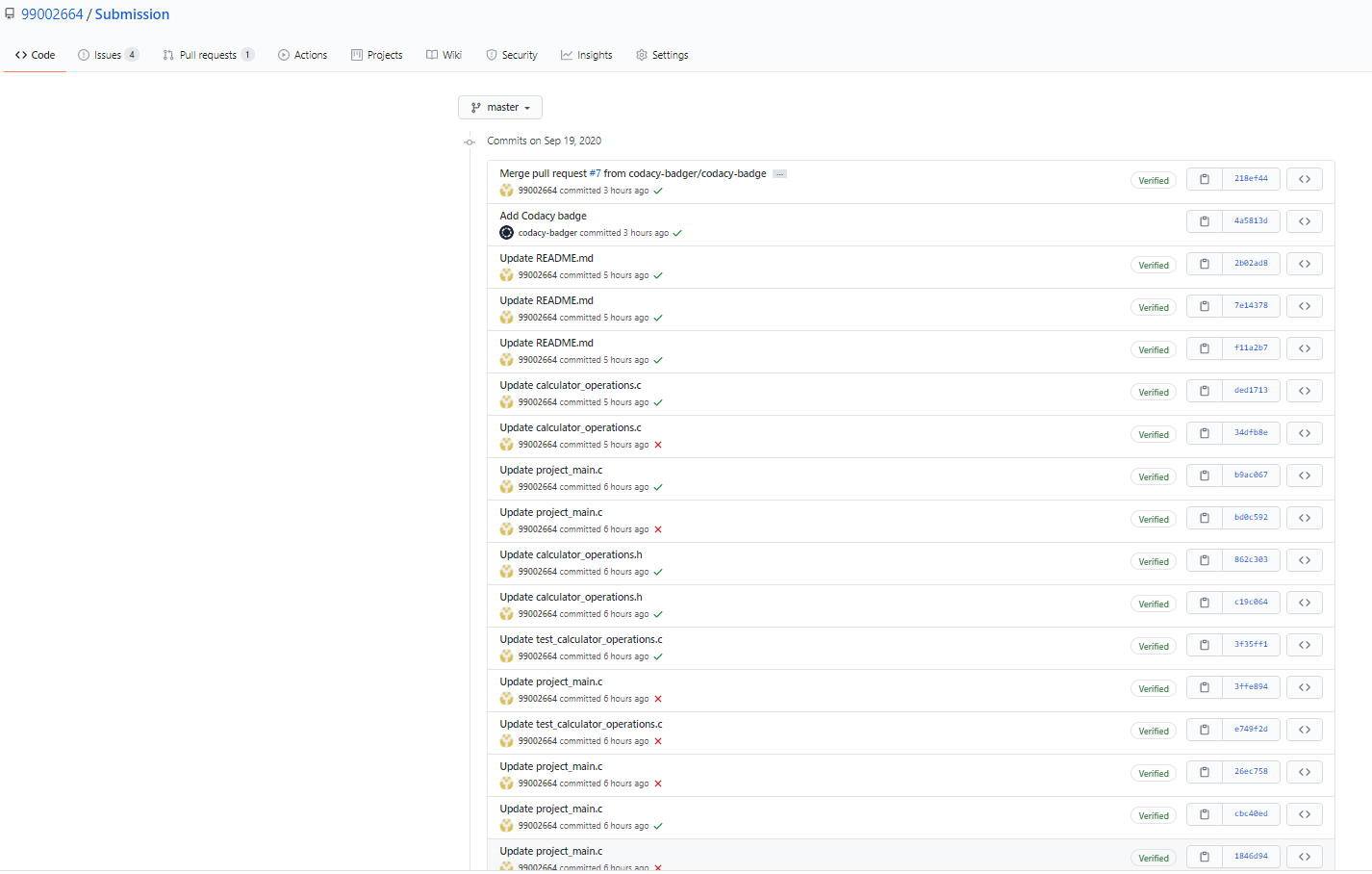


Figure 14: Commits made in GIT repo

### 3.4.2 Make and Build

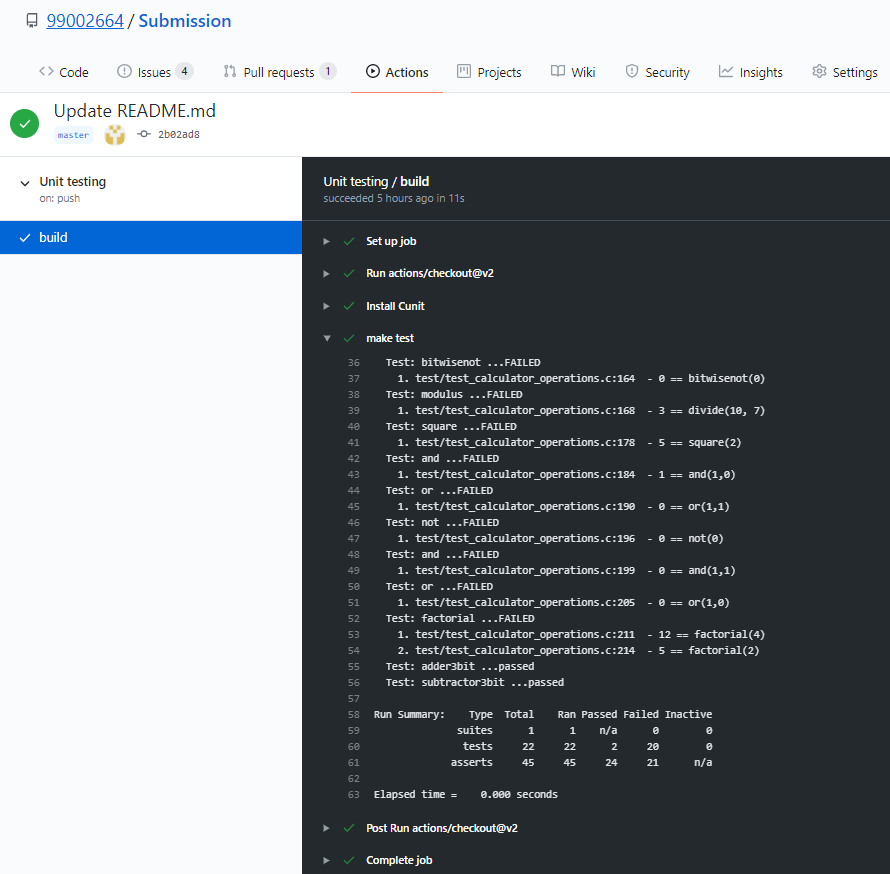


Figure 15: Unit testing build

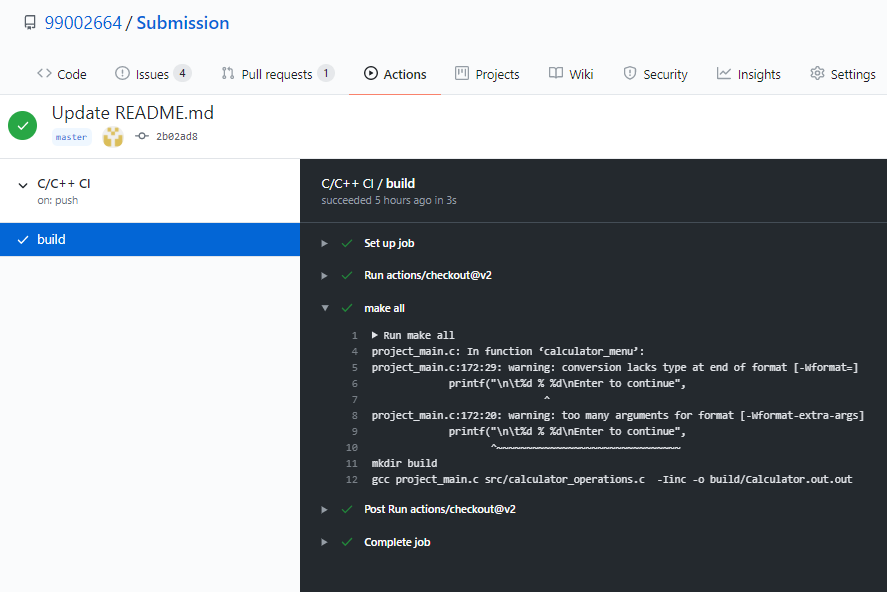


Figure 16: Make file

### 3.4.3 Code Quality

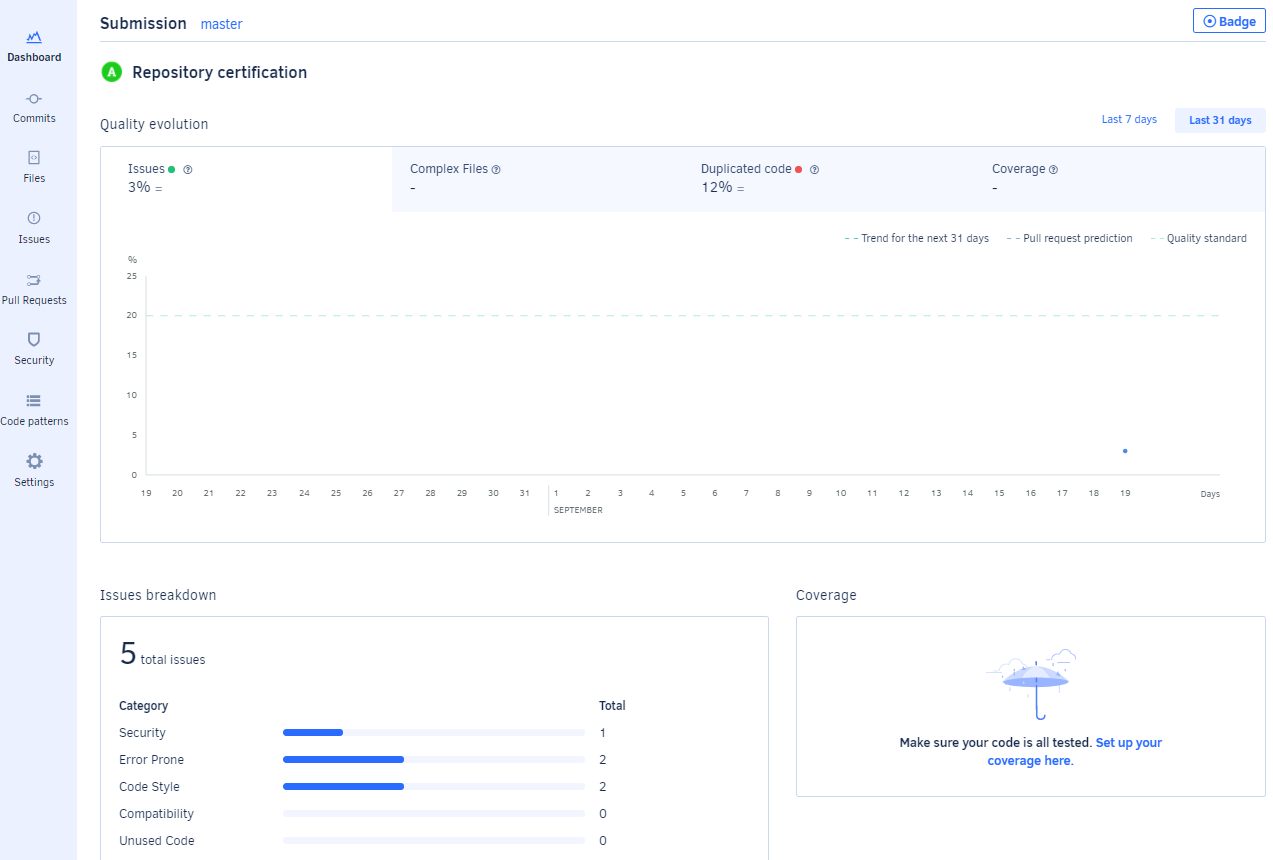


Figure 17: Code quality report

github link: <https://github.com/99002664/Submission>

# APPENDIX

## 1. C program on GIT

|  |
| --- |
| #include "Prime.h" |
|  | /\* |
|  | Calculates whether the given number is prime or not |
|  | bool: used show the result in TRUE or FALSE state |
|  | \*/ |
|  | bool prime (int n) { |
|  | if(n<=1) |
|  | { |
|  | return false; |
|  | } |
|  | For (int i=2 ; i<n/2 ; i++) |
|  | { |
|  | if(n%i==0) |
|  | { |
|  | return false; |
|  | } |
|  | } |
|  | return true; |
|  | } |

(link to github: <https://github.com/stepin105258/Calculator>)

## 2. CI for C program of STEPin module

### 2.1 GIT

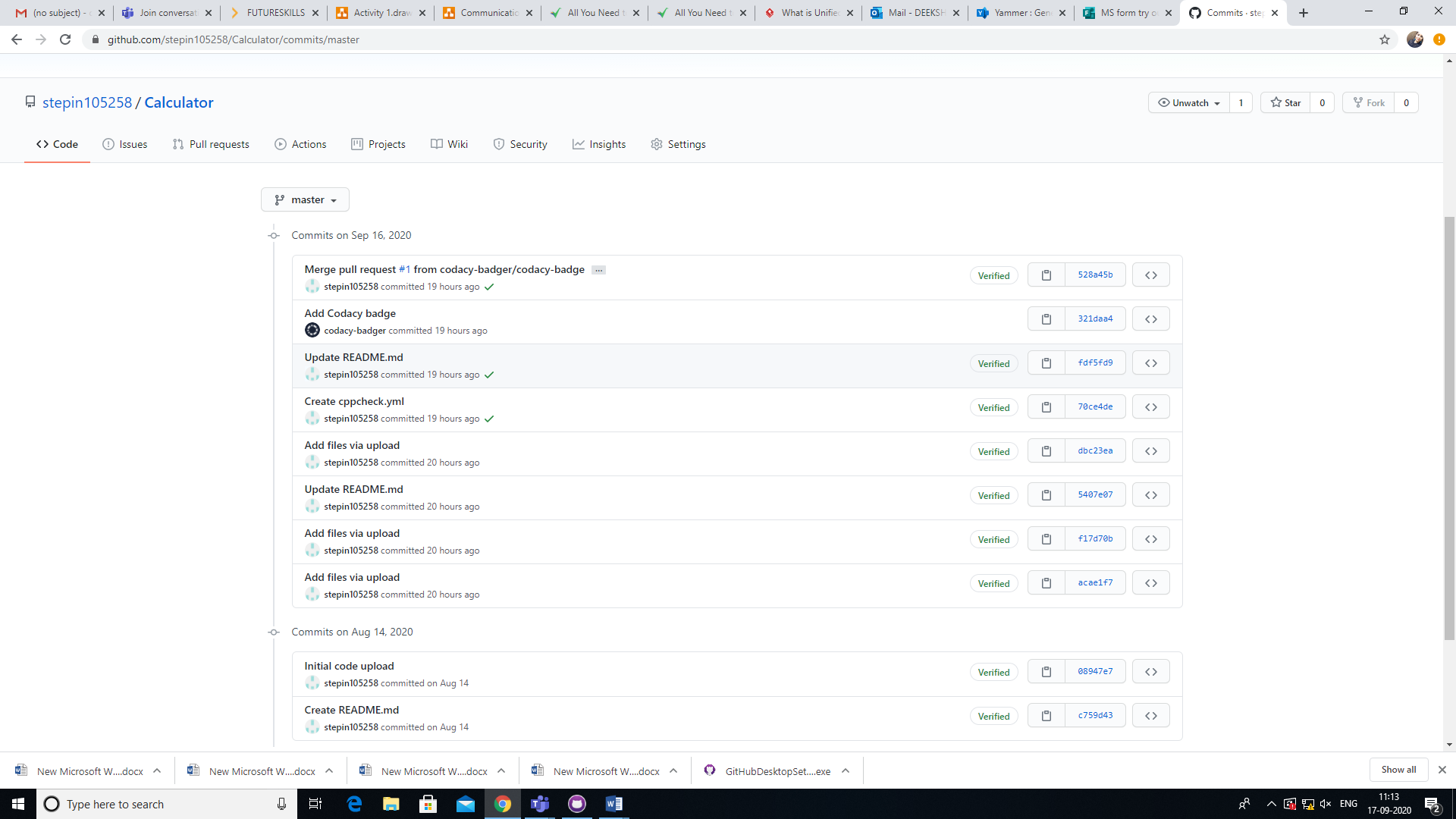


Figure 18: Commits made

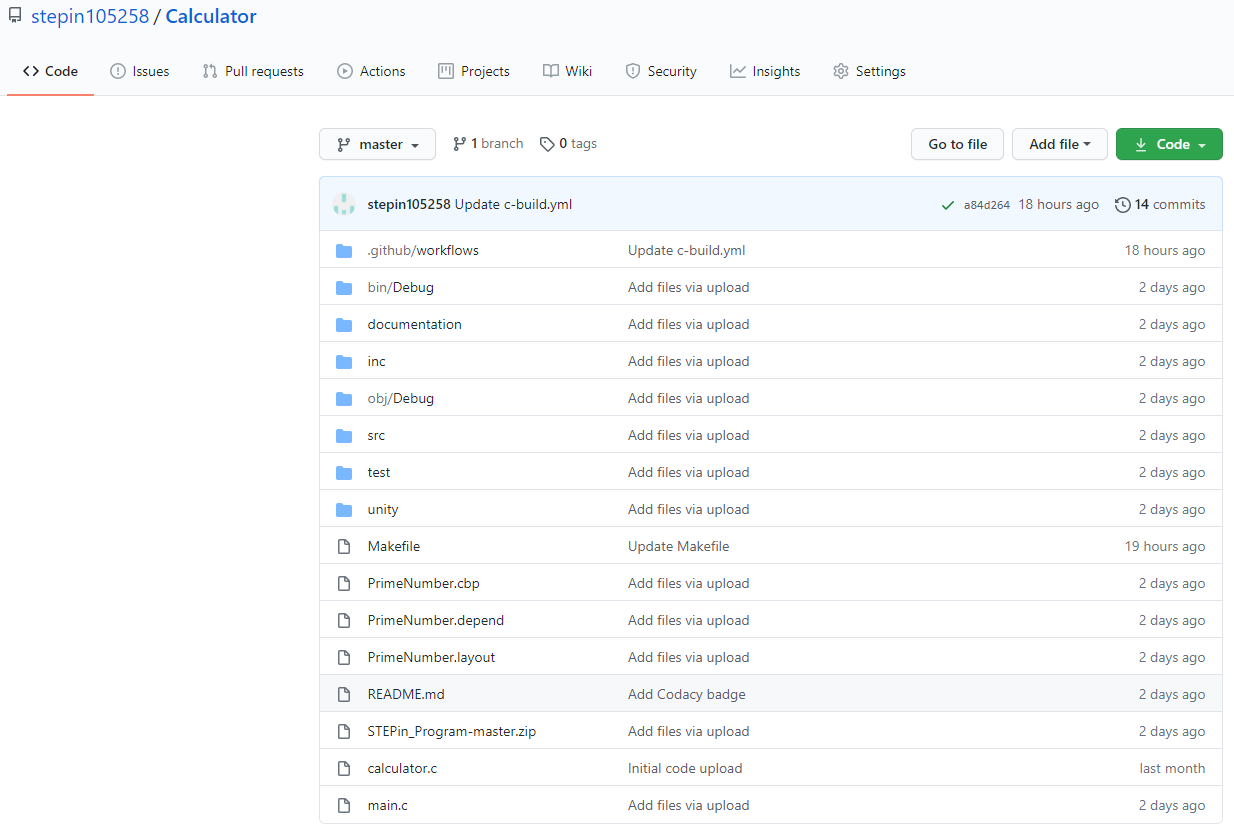


Figure 19: Folders in GIT repo

### 2.2 Make and Build

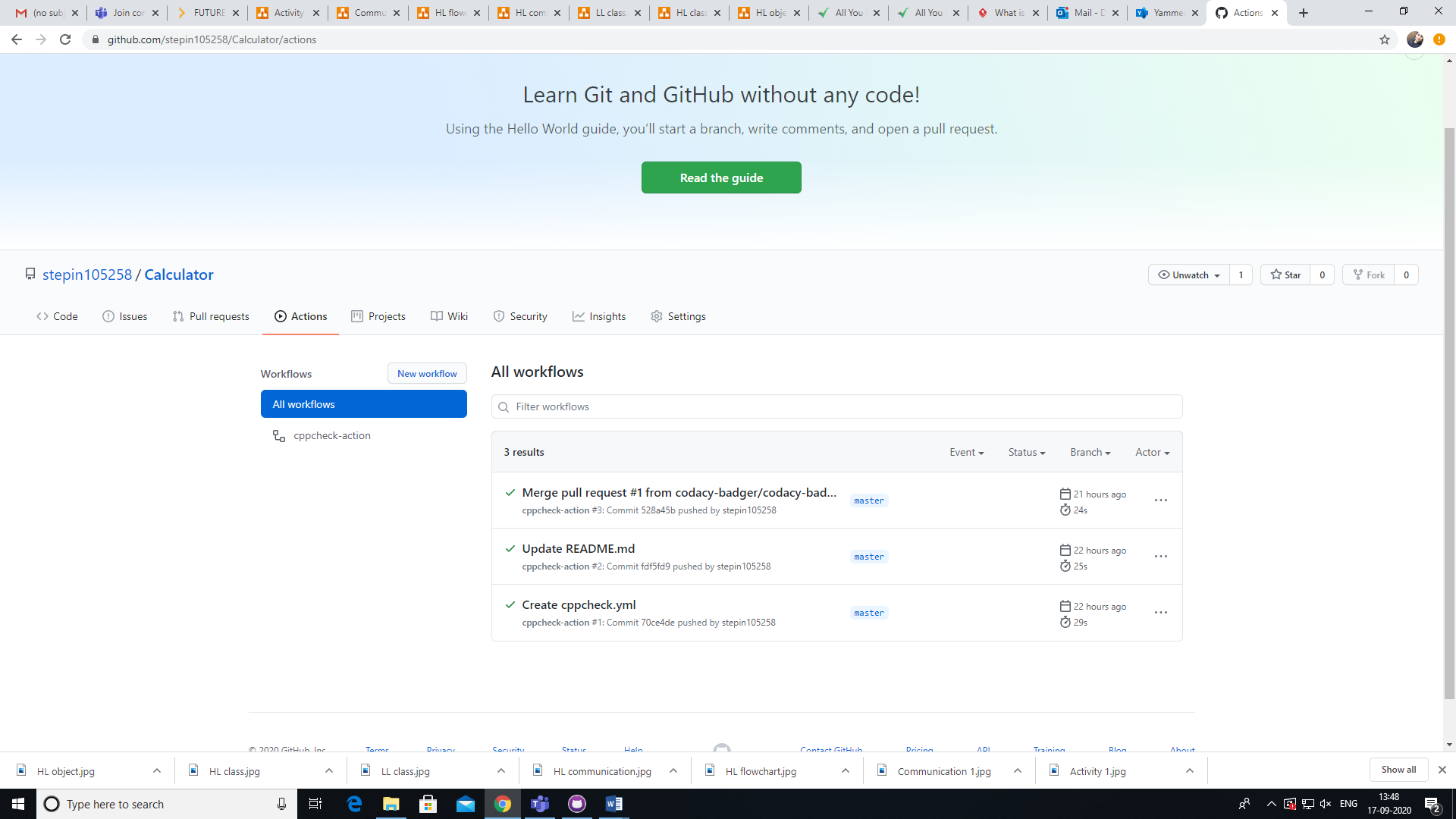


Figure 20: Screenshot of build

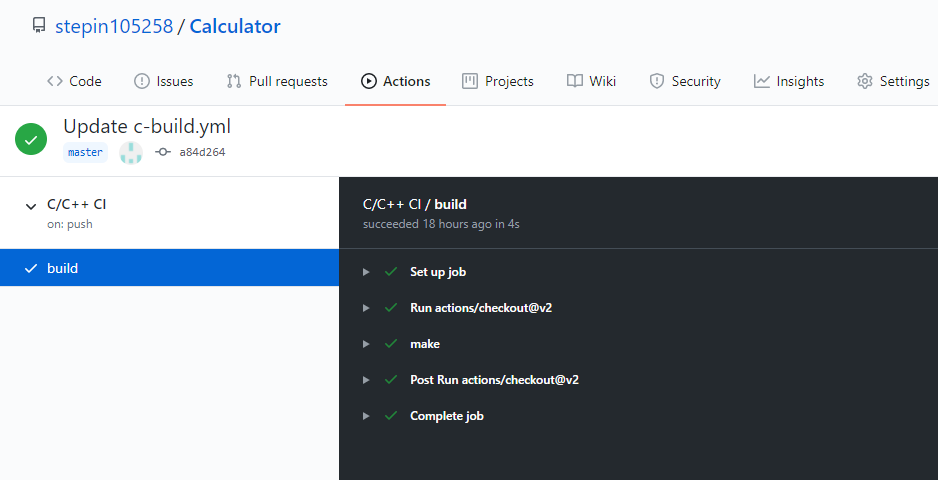


Figure 21: Make file

## 2.3 Code Quality

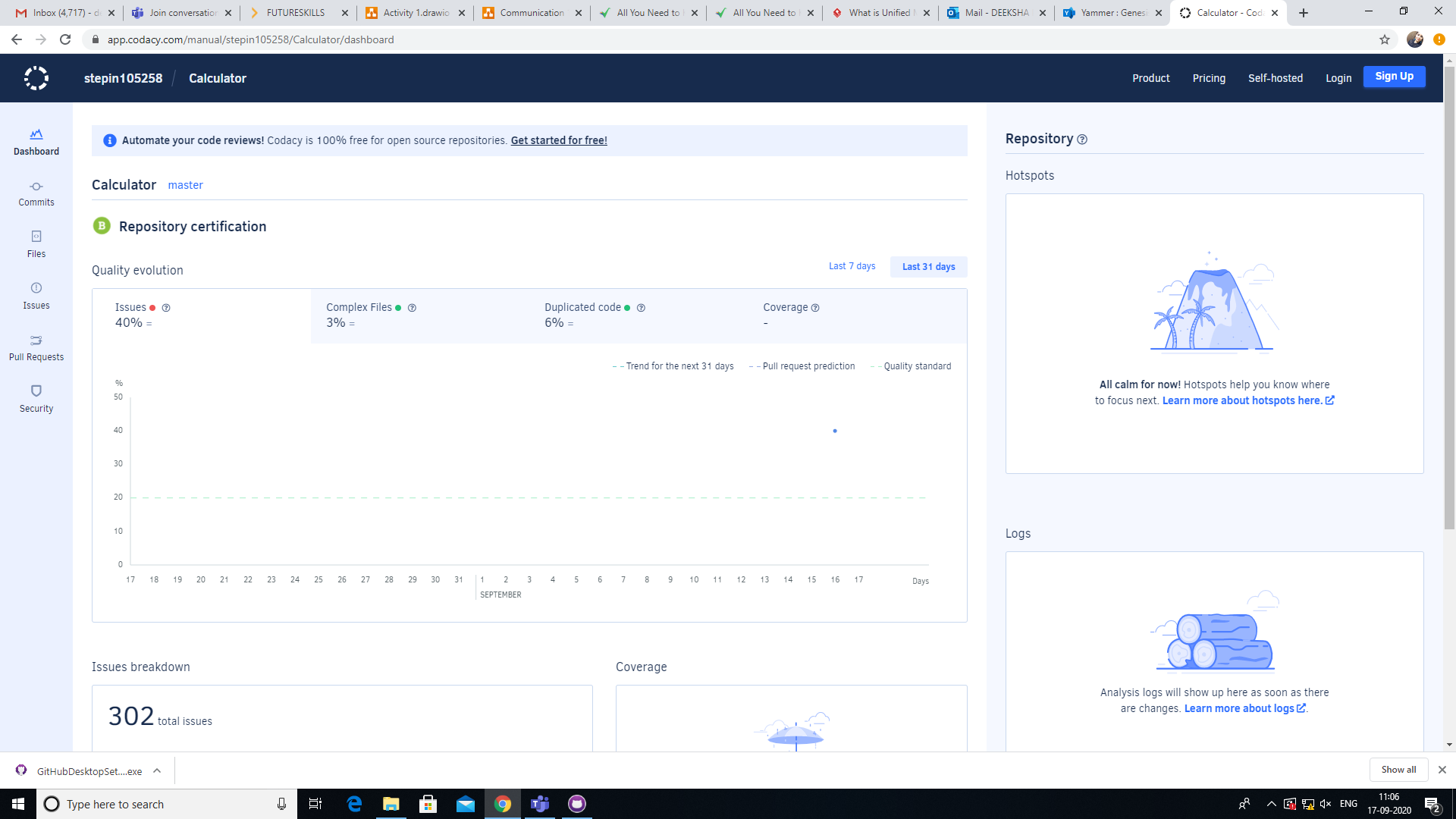


Figure 22: Code quality

# REFERENCES

[1] AnkitaAgarwal, “Secret Key Encryption Algorithm Using Genetic Algorithm”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 4, April 2012

[2] Mustafa Emad Hameed1,2, Masrullizam Mat Ibrahim1 , NurulfajarAbdManap1, “Review on Improvement of Advanced Encryption Standard (AES) Algorithm based on Time Execution, Differential Cryptanalysis and Level of Security”, Journal of Telecommunication, Electronic and Computer Engineering, Jan 2018

[3] AmitChatterjee, JitendraDhanotia, Vimal Bhatia, SantoshRana, ShashiPrakash, “Optical Image Encryption using Fringe Projection Profilometry, Fourier Fringe Analysis, and RSA Algorithm”, 978-1-5386-4318-1/17/$31.00 ©2017 IEEE.

[5] SandeepSrivastava, Sanjay Kumar, “Image Encryption using Simplified Data Encryption Standard (S-DES)”, International Journal of Computer Applications (0975 – 8887) Volume 104 – No.2, October 2014.

[6] Anup&Suchithra, “Image Encryption using Triple DES Algorithm”, Imperial Journal of Interdisciplinary Reasearch (IJIR), Vol-3, Issue-5, 2017.

[7] JianchengZou , Rabab K. Ward , Dongxu Qi, “A New Digital Image Scrambling Method Based on Fibonacci Number, “Proceeding of the IEEE Inter Symposium On Circuits and Systems, Vancouver ,Canada ,Vol .03 , PP .965-968 , 2004.

[8] GuoshengGu ,Guoqiang Han, “An Enhanced Chaos Based Image Encryption Algorithm”,

[9] RiahUkurGinting, Rocky YefrenesDillak, “Digital Color Image Encryption Using RC4 Stream Cipher and Chaotic Logistic Map” @2013 IEEE

[10] N Singh, A Singh- Optics and Lasers in Engineering, “Optical image encryption using fractional Fourier transform and chaos”, Elsevier 2008

[11] [www.academia.com](http://www.academia.com)

[12] [www.draw.io](http://www.draw.io)