# Learning Report – Applied SDLC and Software Testing

GLOBAL ENGINEERING ACADEMY

GENSIS

L&T Technology Services

**Document History**

| Ver. Rel. No. | Release Date | Prepared. By | Reviewed By | Approved By | Remarks/Revision Details |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

**Table of Contents**

Activity and Tasks

**Task 1**– System/ Software Development

**SMART LOCK FOR YOUR SECURITY AND SAFETY**

**RESEARCH**

**Ageing and Cost grading**

| AGEING | COSTING |
|---|---|
| 1) Traditional Lock and key | 500-1500 |
| 2) Combination Lock | 1000-3000 |
| 3) Electronic key card lock | 8000 |
| 4) PIN secured Lock | 15000 |
| 5) Biometrics Locking system | 20000 |

**DEFINITION OF PRODUCT**

1) Wi-Fi enabled remote access from anywhere using smart phones. This feature provides additional safety as you do not have to touch the door to open it in the current COVID scenario.
2) RFID card locking system is also provided. So that the card can be scanned to unlock the system as well.
3) Fingerprint enabled locking system: The fingerprint of the people accessing the building can be stored within the system so that the access can be restricted to a limited group.
4) Remotely knowing when any person is accessing the lock. This feature provides additional security to the door.

**SWOT ANALYSIS**

| | |
|---|---|
| STRENGTH<br><br>1) Contact less access to the building<br>2) Remote monitoring of the people accessing the door<br>3) Remote control of the door<br>4) Safety and security is ensured<br>5) Limited access to the building | WEAKNESS<br><br>1) High cost of the system<br>2) Wi-Fi connection may get disconnected |
| OPPORTUNITIES<br><br>1) High security buildings<br>2) Office buildings to remotely access the people entering the building<br>3) Residential apartments when people forget to lock the door while stepping out<br>4) Contact less accessing of doors in current COVID situation. | THREATS<br><br>1) Hacking the system may lead to vulnerable data being exposed<br>2) Missing of RFID tags can lead to external people accessing the building |

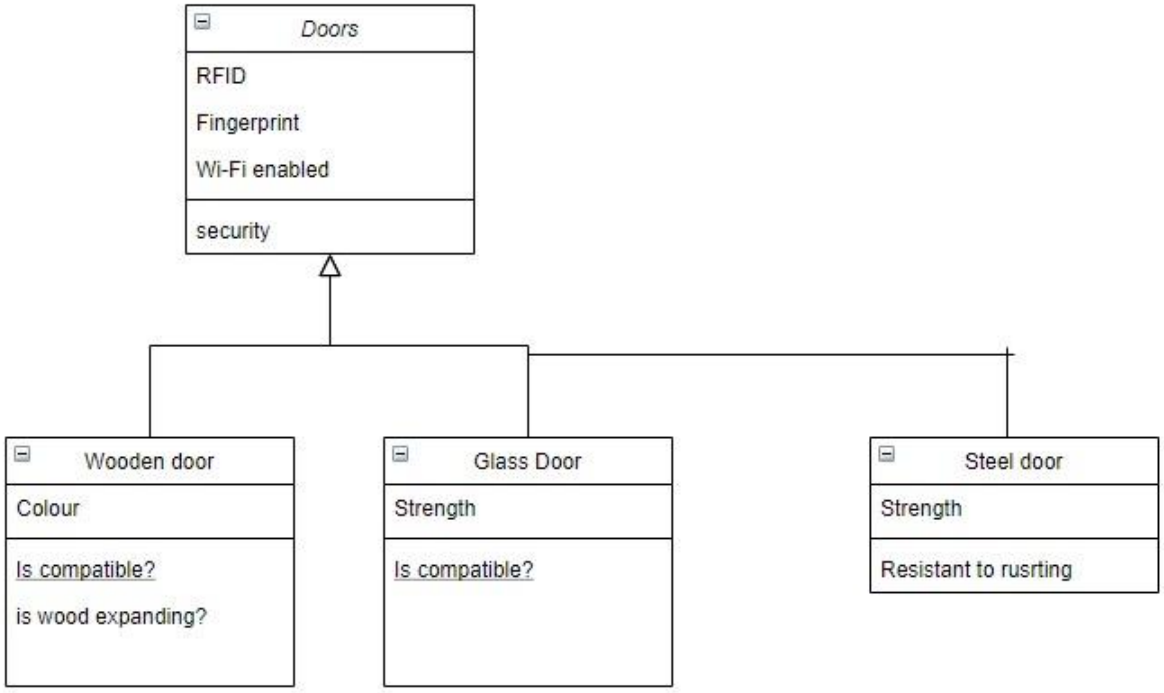
**REQUIREMENTS**

| ID | DESCRIPTION |
|---|---|
| **HL_01** | Wi-Fi enabled remote access to the building to ensure safety of people using the doors in current COVID situation |
| **HL_02** | Accessing the building using RFID tag |
| **HL_03** | Fingerprint access to building to limit the people accessing the building and thereby ensure security. |
| **HL_04** | Material Quality |
| **HL_05** | Space Usability |
| **HL_06** | Aesthetically pleasing |
| **HL_01_LL_01** | Wi-Fi Connectivity |
| **HL_02_LL_01** | Power source management |
| **HL_03_LL_01** | Database Management |
| **HL_04_LL_01** | Resistant to corrosion |
| **HL_04_LL_02** | Resistant to different weather conditions |
| **HL_05_LL_01** | Light weight device |
| **HL_05_LL_02** | Easy to manage and handle |

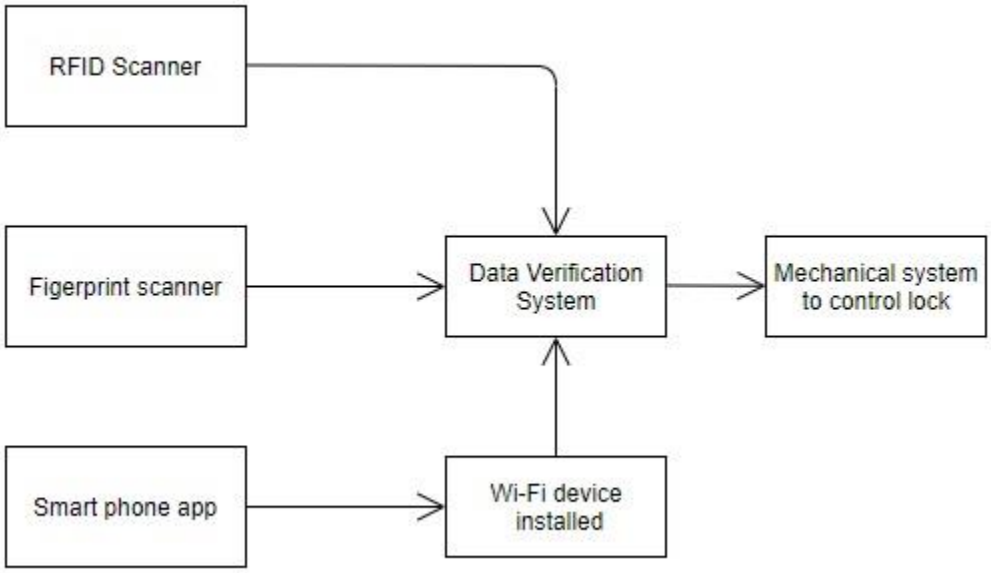
**SYSTEM DESIGN**

## 1) Structural Diagram

- CLASS DIAGRAM

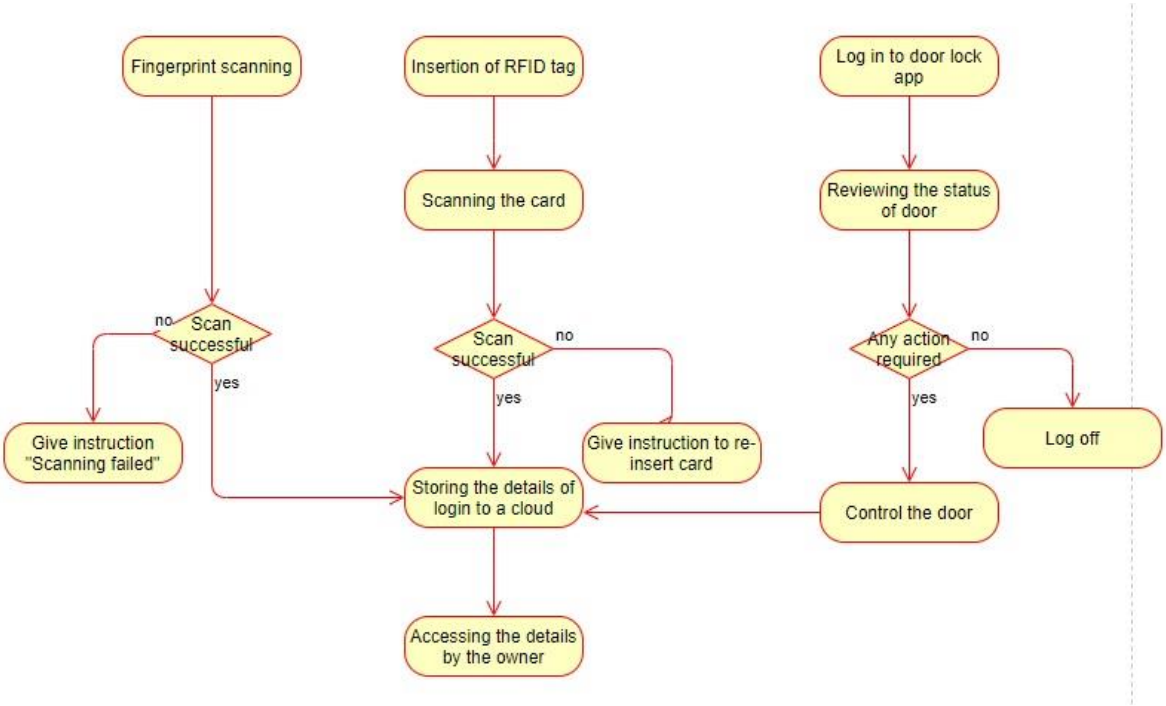


- COMPONENT DIAGRAM

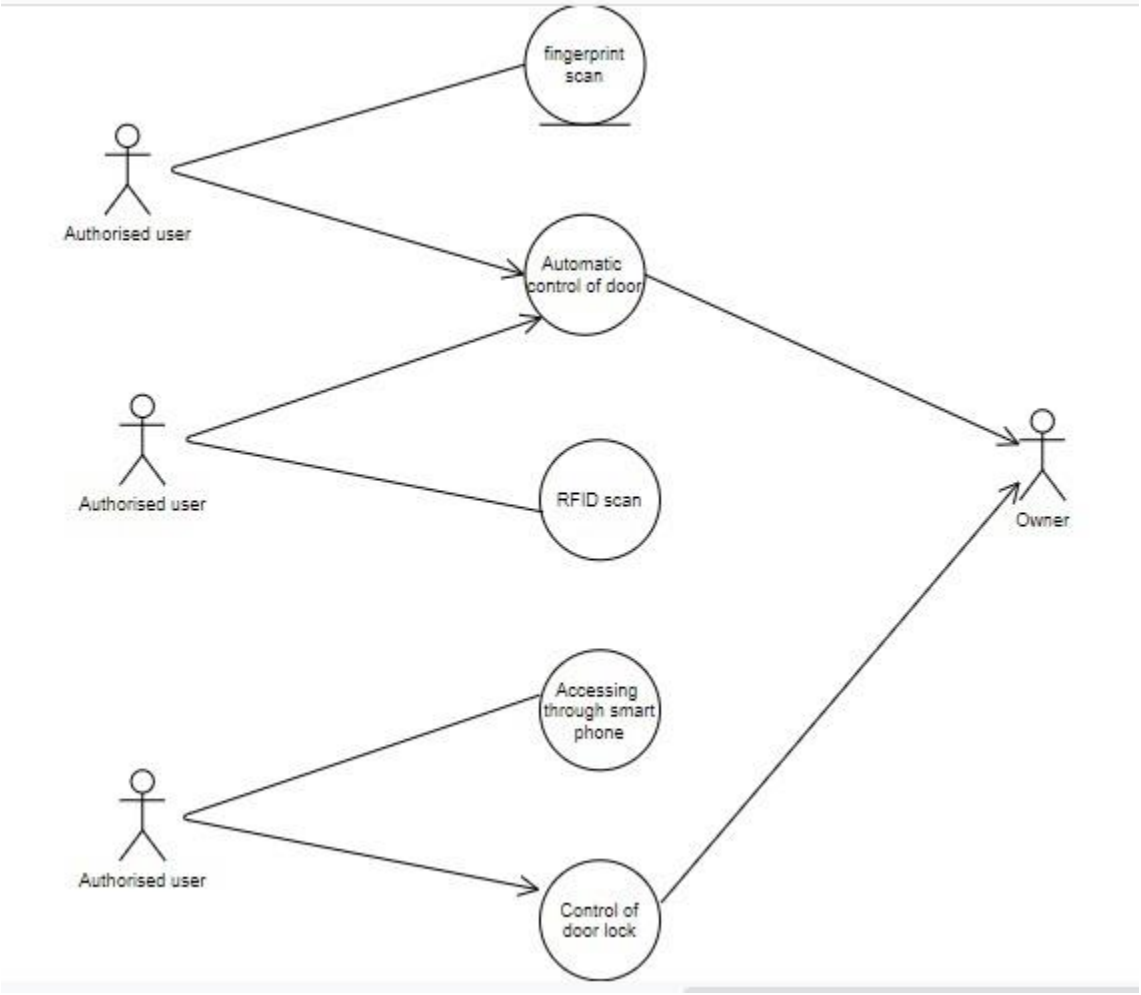


## 2) Behavioral Diagram

- ACTIVITY DIAGRAM



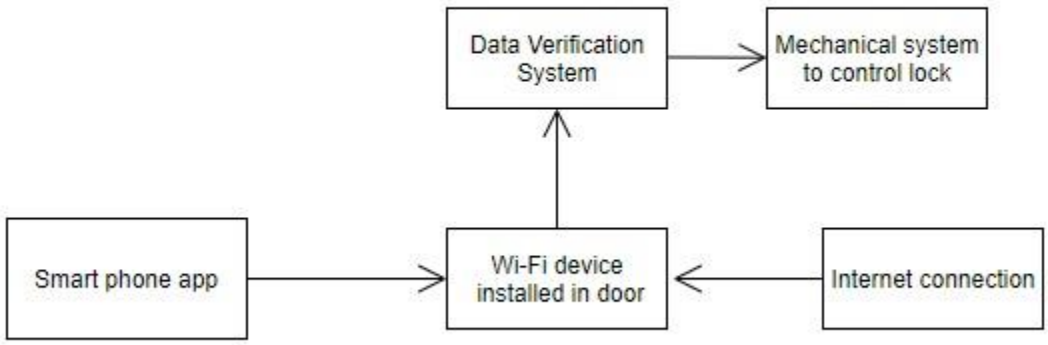
- USECASE DIAGRAM



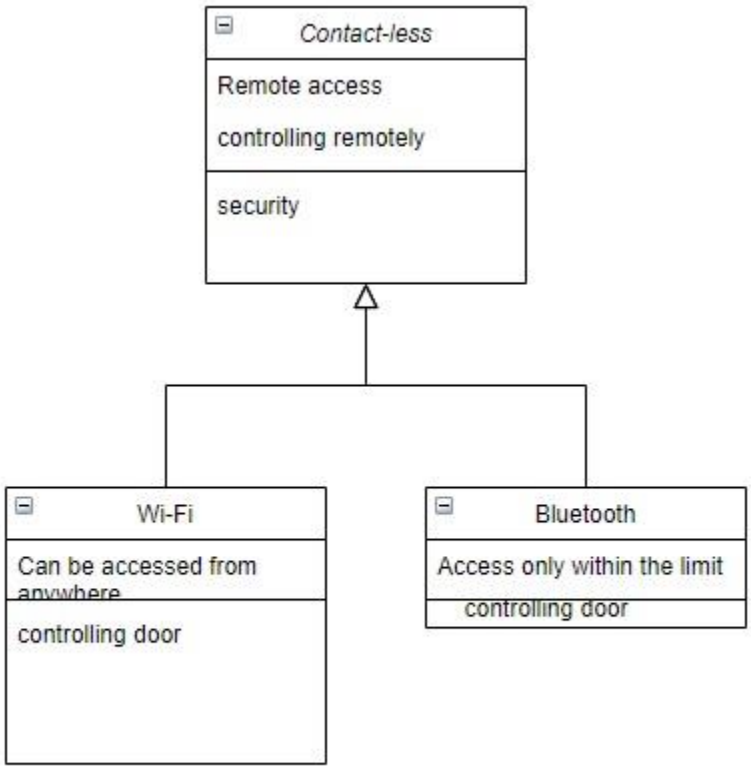
**MODULAR DESIGN**

**1) Structural Diagram**

- COMPONENT DIAGRAM

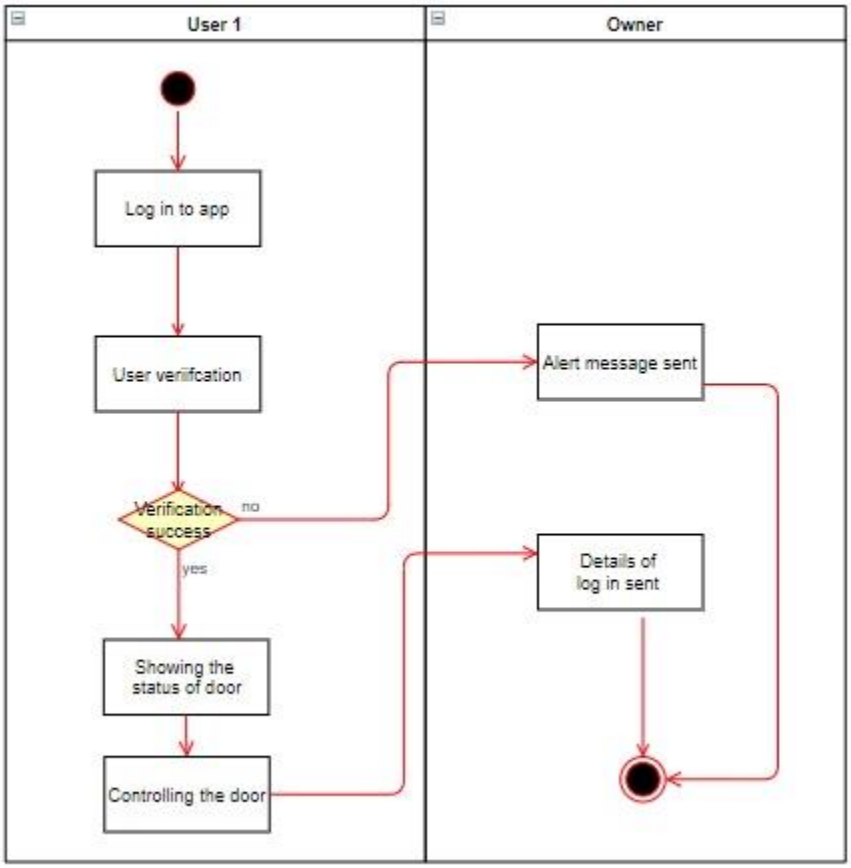


- CLASS DIAGRAM

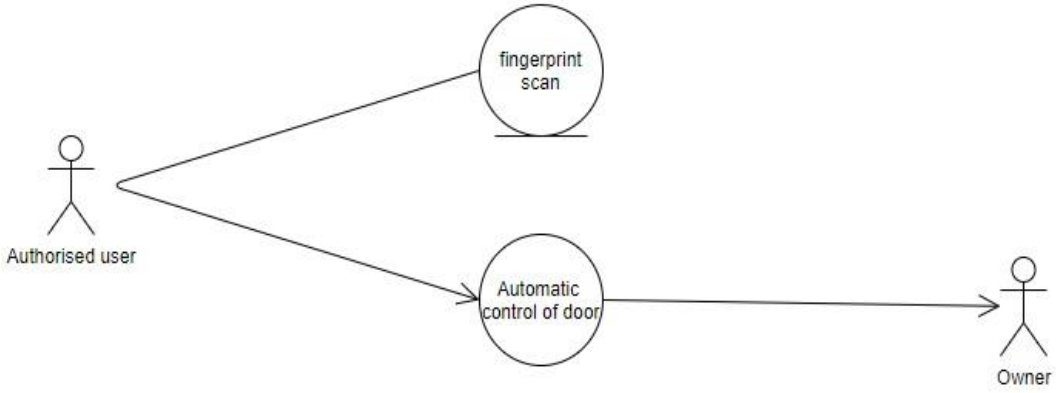


## 2) Behavioral Diagram

- ACTIVITY DIAGRAM



- USECASE DIAGRAM




**TEST PLAN**

**Integration Test Plan and Unit Test Plan**

| ID | DESCRIPTION | PRE-CONDITION | EXPECTED INPUT | EXPECTED OUTPUT | ACTUAL OUTPUT |
|---|---|---|---|---|---|
| **HL_01_IT_01** | Testing all the features of Wi-Fi enabled device using a smart phone App for its proper functioning | No Wi-Fi connection | Log in to device app using credentials and access the features | Proper functioning of features | |
| **HL_02_IT_01** | Testing the RFID tags for its functions | RFID tags may not be received | Inserting the RFID tag and scanning the tag | Scan successful and door opens | |
| **HL_03_IT_01** | Testing the fingerprint access of an authorized person for successful scan | Fingerprint may not be stored in database | Fingerprint scan | Scan successful | |
| **HL_04_IT_01** | Testing against different industry standards | System may not be properly integrated | Different materials used in the system | All the materials meet the expected standards | |
| **HL_05_IT_01** | Testing for the space used up by the device and compare it with the acceptable | System may not be integrated | System | The space taken up by the system is less than 5% of the door | |
| **HL_06_IT_01** | Checking whether the appearance of the system is acceptable in the market and will obtain good market value | System may not be working | System | The product will have good market value which can be expected to get required profits | |

| | | | | | |
|---|---|---|---|---|---|
| **HL_01_LL_01_UT_01** | Testing the Wi-Fi connectivity of the device | The Wi-Fi may not be turned on | Wi-Fi | A good and stable connectivity is ensured | |
| **HL_02_LL_01_UT_01** | Testing the battery life of the system | The battery may not be charged | Power source | A good battery life is ensured | |
| **HL_03_LL_01_UT_01** | Testing for the proper management and quality of database of the system | The database may not be completed | Database of the system | A well-managed database | |
| **HL_04_LL_01_UT_01** | Testing against different corroding agents | A well painted system | Any corroding agent | The product is resistant against the inputs | |
| **HL_04_LL_02_UT_01** | Testing against the weather conditions that are probable to occur | System may not be integrated | Different weather conditions | The product is resistant against the inputs | |
| **HL_05_LL_01_UT_01** | The weight of the system is checked and compared against the overall weight of the door | System may not be integrated | Weight | The product is light weight and the weight of the product is less than 5% of the door | |
| **HL_05_LL_02_UT_01** | The overall functionality and ease of handling is monitored and compared with existing devices | System may not be integrated | System | Comfortable and easy to handle | |

**1) Requirement Based**

- Testing the features of the system using the smart phone apps.
- Testing the RFID tags for its functions.
- Testing the fingerprint access of the system
- Material quality testing.
- Testing the space usability.
- Testing the battery life

**2) Scenario Based**

- Testing against different weather conditions.
- Testing the system for protection against corrosion
- Testing to understand any situation of hacking the system

**3) Boundary conditions**

- Testing the system when there is a Wi-Fi connectivity issue.
- Testing when the RFID tags are lost
- Testing to detect the functioning of the system when an external person who do not have access to the building tries to enter into the building
- Testing to find any situation of over-heating and exploding of battery

**REFERENCES**


- ***Smart lock buying guide CNET***, [online] Available: https://www.cnet.com/news/smart-lock-buving-guide/.
- A. Kassem, S. E. Murr, G. Jamous, E. Saad and M. Geagea, "A smart lock system using Wi-Fi security", ***2016 3rd International Conference on Advances in Computational Tools for Engineering Applications (ACTEA)***, pp. 222-225, 2016.
- Yu Yuan, D. Shing-chern and Dwen-Ren Tsai, "Smart Door: A Ubiquitous Collaboration System for Home Activities in the Smart Home", ***J. Inf. Sci. Eng.***, vol. 29, no. 6, pp. 1227-1248, 2013.

**Task 2** – Agile Aspects

**AGILE MODEL**

**1) Theme**

A safe and secured Door lock system that can provide remote access to the lock from anywhere with additional features like limiting the number of people entering the building and maintaining a record of the people accessing the building.

**2) Epic**

- Security is the prime concern/requirement
- Safety must be ensured
- Easily and remote access to people entering the building
- Non-contact controlling of the locking system
- Light weight locking system
- Easy to handle
- Tolerance to extreme weather conditions.
- Resistant to corrosion
- Aesthetically pleasing
- Take up only less space

**3) User Stories**

| ID | Description |
|---|---|
| **US_01** | As a user, I am expecting a safe system to use in the current situation of COVID. So that there is minimal requirement to open/close the door by contact. Also the security of the system has to be taken as a prime concern. I am delivery within 6 months. The system needs to be tested for faster performance and safety. |
| **US_02** | As a user, I am expecting a safe system that can be accessed remotely from anywhere. So that even if I forget to close the door, it can be done remotely. Along with this feature, I am expecting complete safety for this device. I am expecting a delivery within 5 months. The system needs to be tested for any kind of hacking and hecnce security needs to be ensured. |

| | |
|---|---|
| **US_03** | As users, We are expecting a system where the details of the people entering the building can be accessed remotely using a Mobile phone. Also, a remote controlling of the door needs to be provided for security purpose. We are expecting a delivery within 7 months. The product has to be tested for its functionalities and performance. |

**REFERENCES**


- https://www.agilemarketing.net/epic-vs-theme-2/
- https://www.geeksforgeeks.org/software-engineering-agile-development-models/#:~:text=The%20Agile%20model%20adopts%20Iterative,a%20couple%20of%20weeks%20only.&text=Agile%20model%20is%20the%20combination%20of%20iterative%20and%20incremental%20process%20models.
- https://www.yodiz.com/blog/what-is-epic-in-agile-methodology-definition-and-template-of-epic/#:~:text=Epic%20Definition%20in%20Agile%20Scrum,customer%20request%20or%20business%20requirement.&text=These%20details%20are%20defined%20in,than%20one%20sprint%20to%20complete.

# TASK 3 – MINIPROJECT

## PROPERTY ANALYZER OF ANY GIVEN NUMBER

### INTRODUCTION

This is an application which checks for various properties of a number. When a number is given, the application checks whether the number is Prime, Armstrong, Odd/Even, Palindrome, Power, Harshad, Perfect_square, Perfect_cube, Automorphic, Divisible by 3, Divisible by 5 and Divisible by 7.

### REQUIREMENTS

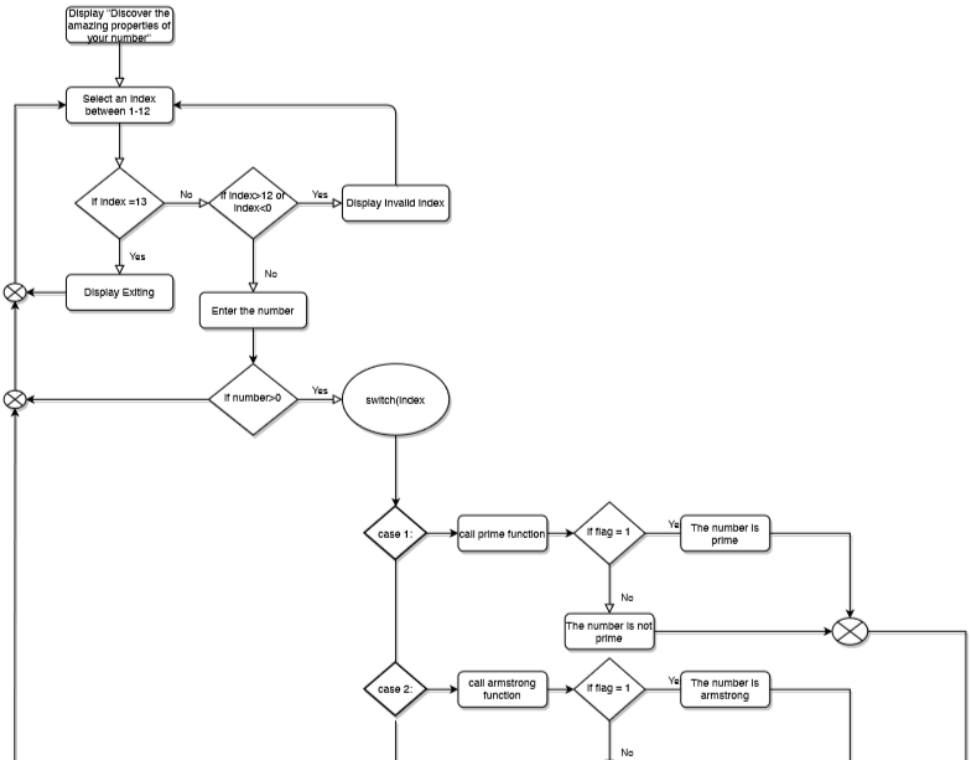
#### HIGH LEVEL REQUIREMENTS

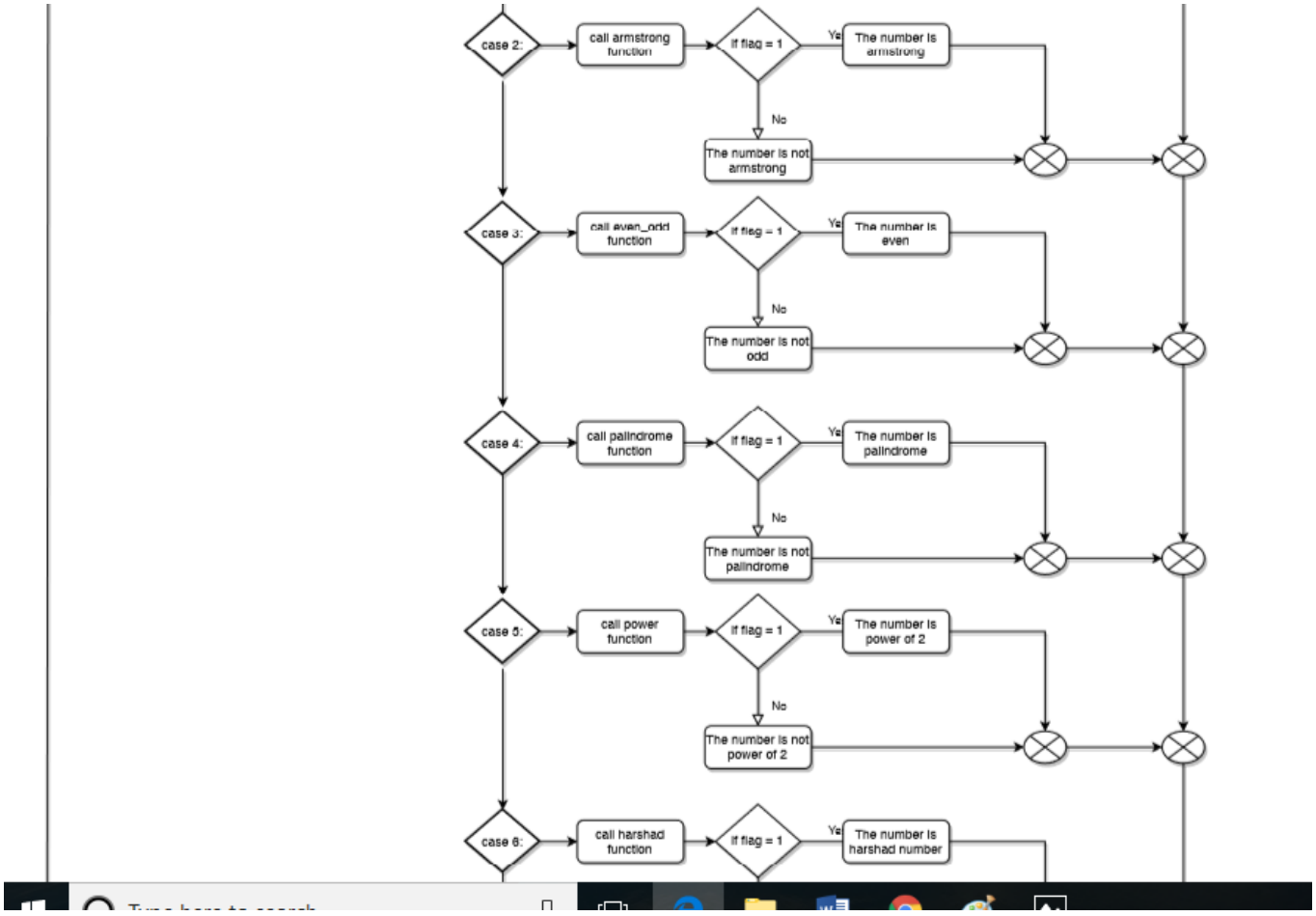
| ID | DESCRIPTION |
|---|---|
| **HL_01** | Check for different properties of the number |
| **HL_02** | Determine the divisibility |
| **HL_03** | Perform different operations |

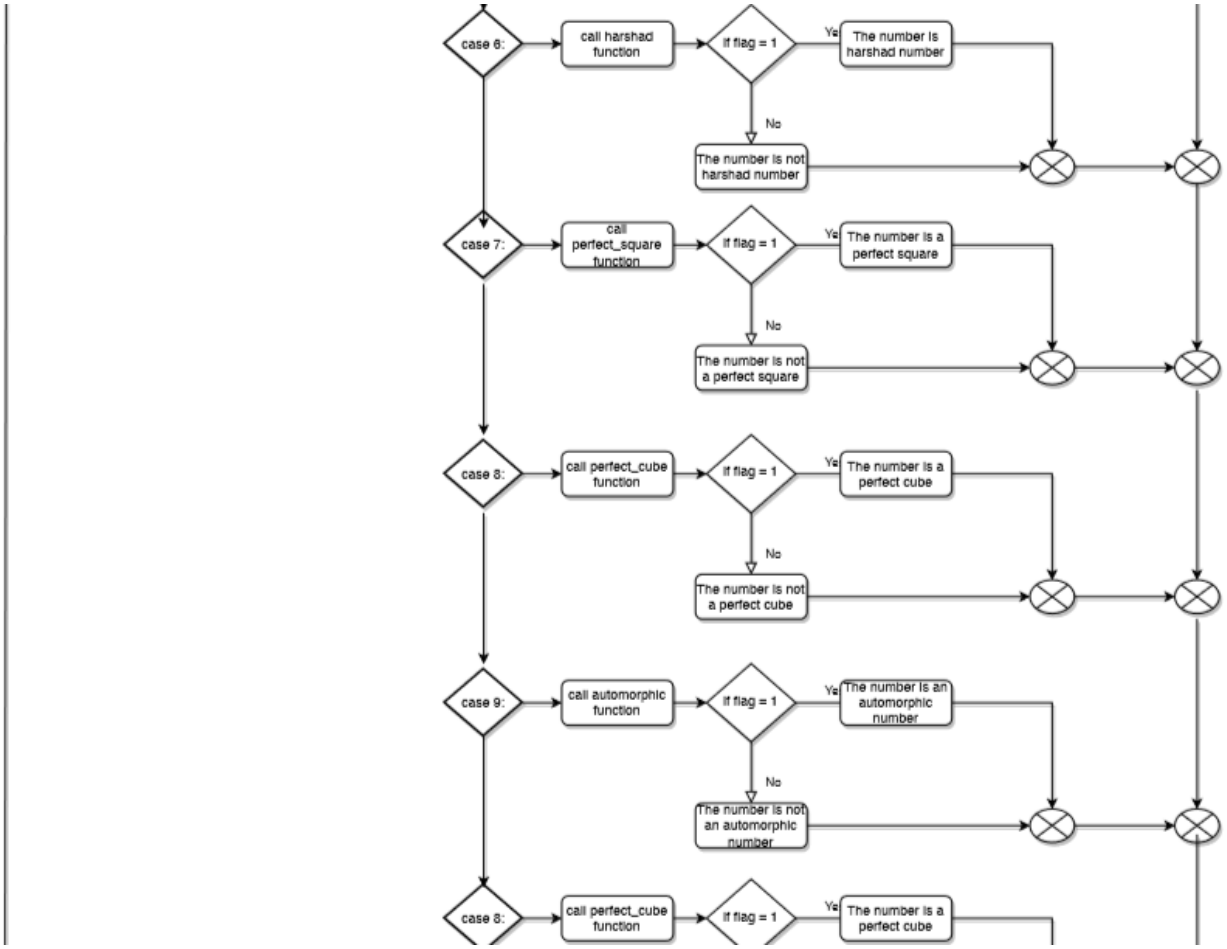
**LOW LEVEL REQUIREMENTS**

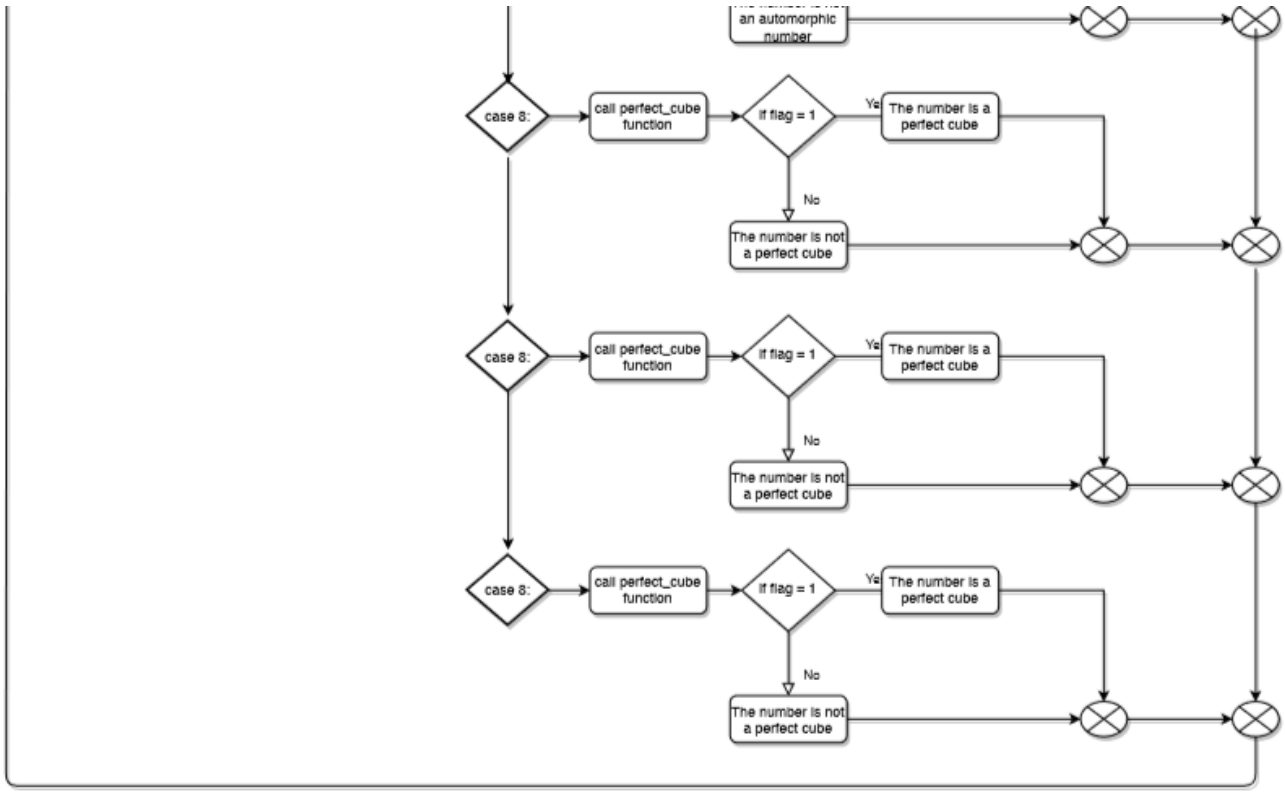
| ID | DESCRIPTION |
|---|---|
| **HL_01_LL_01** | Determine whether the given number is prime |
| **HL_01_LL_02** | Determine whether the given number is Armstrong |
| **HL_01_LL_03** | Determine whether the given number is Odd or Even |
| **HL_01_LL_04** | Determine whether the given number is a Palindrome |
| **HL_01_LL_05** | Determine whether the given number is a power of 2 |
| **HL_01_LL_06** | Determine whether the given number is a Harshad number |
| **HL_01_LL_ 07** | Determine whether the given number is a perfect square |
| **HL_01_LL_08** | Determine whether the given number is a perfect cube |
| **HL_01_LL_09** | Determine whether the given number is an Automorphic number |
| **HL_02_LL_01** | Determine whether the given number is divisible by 3 |
| **HL_02_LL_02** | Determine whether the given number is divisible by 5 |
| **HL_02_LL_03** | Determine whether the given number is divisible by 7 |
| **HL_03_LL_01** | Perform addition operation |
| **HL_03_LL_02** | Perform subtraction operation |
| **HL_03_LL_03** | Perform multiplication operation |
| **HL_03_LL_04** | Perform division operation |

**UML DIAGRAM**




case 2:
call armstrong function
if flag = 1
Ye
The number is armstrong
No
The number is not armstrong
case 3:
call even_odd function
if flag = 1
Ye
The number is even
No
The number is not odd
case 4:
call palindrome function
if flag = 1
Ye
The number is palindrome
No
The number is not palindrome
case 5:
call power function
if flag = 1
Ye
The number is power of 2
No
The number is not power of 2
case 6:
call harshad function
if flag = 1
Ye
The number is harshad number


case 6:
call harshad function
if flag = 1
Ye
The number is harshad number
No
The number is not harshad number
case 7:
call perfect_square function
if flag = 1
Ye
The number is a perfect square
No
The number is not a perfect square
case 8:
call perfect_cube function
if flag = 1
Ye
The number is a perfect cube
No
The number is not a perfect cube
case 9:
call automorphic function
if flag = 1
Ye
The number is an automorphic number
No
The number is not an automorphic number
case 8:
call perfect_cube function
if flag = 1
Ye
The number is a perfect cube


an automorphic number
case 8:
call perfect_cube function
If flag = 1
Ye
The number is a perfect cube
No
The number is not a perfect cube
case 8:
call perfect_cube function
If flag = 1
Ye
The number is a perfect cube
No
The number is not a perfect cube
case 8:
call perfect_cube function
If flag = 1
Ye
The number is a perfect cube
No
The number is not a perfect cube

**TEST PLAN**

| ID | DESCRIPTION | PRE-CONDITION | EXPECTED INPUT | EXPECTED OUTPUT | ACTUAL OUTPUT |
|---|---|---|---|---|---|
| **HL_01_IT_01** | Test Different properties of the given number | The number must be positive | Any Number | PASS/FAIL | |
| **HL_02_IT_01** | Test for the divisibility of number | The number must be positive | Any Number | Divisible | |
| **HL_03_IT_01** | Perform different operations | The number must be positive | Any Number | Result of the operation | |
| **HL_01_LL_01_UT_01** | Testing with a prime number | The number must be positive | 1 | The number is prime | |
| **HL_01_LL_01_UT_02** | Testing with a non- prime number | The number must be positive | 48 | The number is not prime | |
| **HL_01_LL_02_UT_01** | Testing with an Armstrong number | The number must be positive | 153 | The number is Armstrong | |
| **HL_01_LL_02_UT_02** | Testing with non-Armstrong number | The number must be positive | 15 | The number is not Armstrong | |
| **HL_01_LL_03_UT_01** | Testing with an even number | The number must be positive | 2 | The number is Even | |
| **HL_01_LL_03_UT_02** | Testing with an odd number | The number must be positive | 5 | The number is Odd | |
| **HL_01_LL_04_UT_01** | Testing with a palindrome number | The number must be positive | 1551 | The number is Palindrome | |

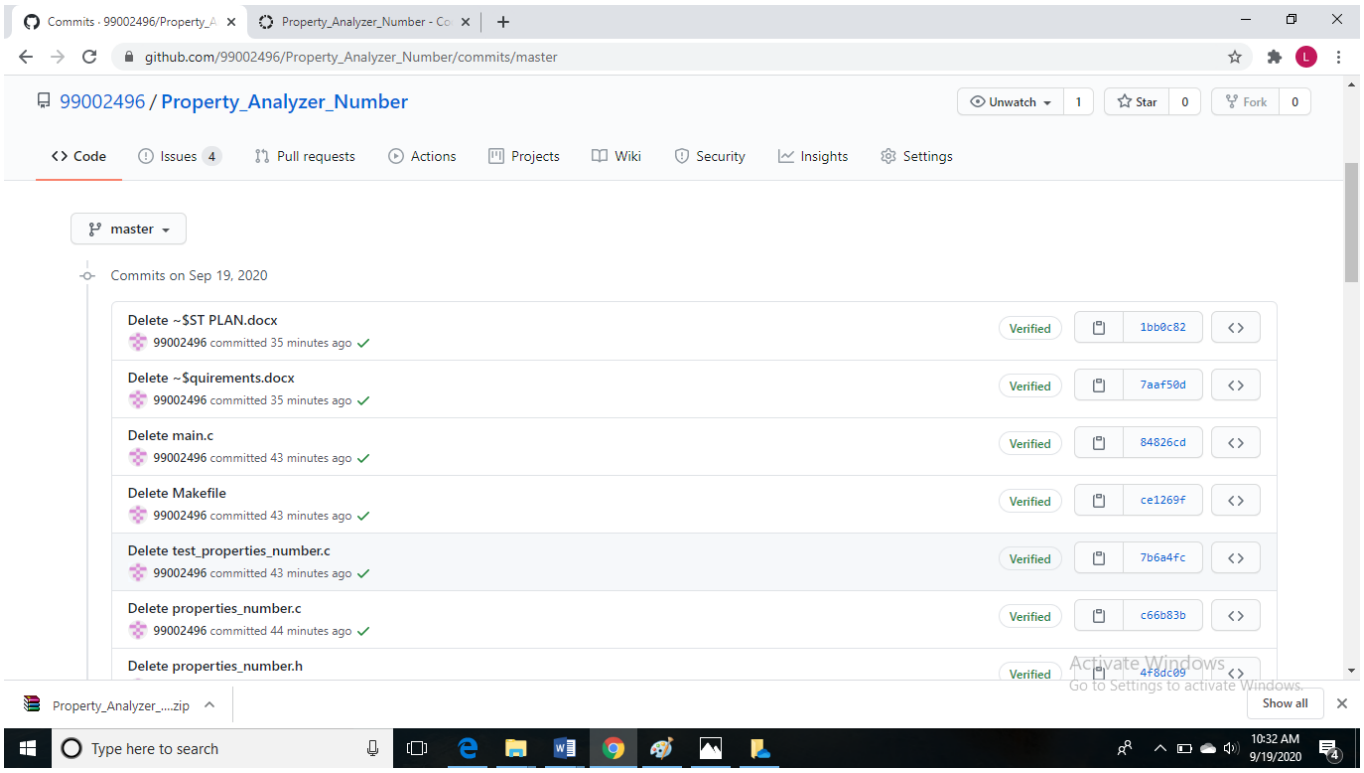
| | | | | | |
|---|---|---|---|---|---|
| **HL_01_LL_04_UT_02** | Testing with a number which is not a palindrome | The number must be positive | 15 | The number is not Palindrome | |
| **HL_01_LL_05_UT_01** | Testing with a number which is a power of 2 | The number must be positive | 4 | The number is a power of 2 | |
| **HL_01_LL_05_UT_02** | Testing with a number which is not a power of 2 | The number must be positive | 5 | The number is not a power of 2 | |
| **HL_01_LL_06_UT_01** | Testing with a Harshad number | The number must be positive | 156 | The number is Harshad | |
| **HL_01_LL_06_UT_02** | Testing with a non- Harshad number | The number must be positive | 15 | The number is not harshad | |
| **HL_01_LL_07_UT_01** | Testing with a perfect square | The number must be positive | 25 | The number is a perfect square | |
| **HL_01_LL_07_UT_02** | Testing with a number which is not a perfect square | The number must be positive | 8 | The number is not a perfect square | |
| **HL_01_LL_08_UT_01** | Testing with a perfect cube | The number must be positive | 8 | The number is a perfect cube | |
| **HL_01_LL_08_UT_02** | Testing with a number which is not a perfect cube | The number must be positive | 5 | The number is not a perfect cube | |
| **HL_01_LL_09_UT_01** | Testing with an automorphic number | The number must be positive | 5 | The number is automorphic | |
| **HL_01_LL_09_UT_02** | Testing with a non-automorphic number | The number must be positive | 78 | The number is not automorphic | |

| | | | | | |
|---|---|---|---|---|---|
| **HL_02_LL_01_UT_01** | Testing with a number divisible by 3 | The number must be positive | 6 | The number is divisible by 3 | |
| **HL_02_LL_01_UT_02** | Testing with a number not divisible by 3 | The number must be positive | 7 | The number is not divisible by 3 | |
| **HL_02_LL_02_UT_01** | Testing with a number divisible by 5 | The number must be positive | 25 | The number is divisible by 5 | |
| **HL_02_LL_02_UT_02** | Testing with a number not divisible by 5 | The number must be positive | 8 | The number is not divisible by 5 | |
| **HL_02_LL_03_UT_01** | Testing with a number divisible by 7 | The number must be positive | 49 | The number is divisible by 7 | |
| **HL_02_LL_03_UT_02** | Testing with a number not divisible by 7 | The number must be positive | 8 | The number is not divisible by 7 | |
| **HL_03_LL_01_UT_01** | Testing Addition | The number must be positive | 15<br>12 | 27 | |
| **HL_03_LL_02_UT_01** | Testing Subtraction | The number must be positive | 48<br>5 | 43 | |
| **HL_03_LL_03_UT_01** | Testing Multiplication | The number must be positive | 5<br>6 | 30 | |
| **HL_03_LL_04_UT_01** | Testing Division | The number must be positive | 25<br>5 | 5 | |

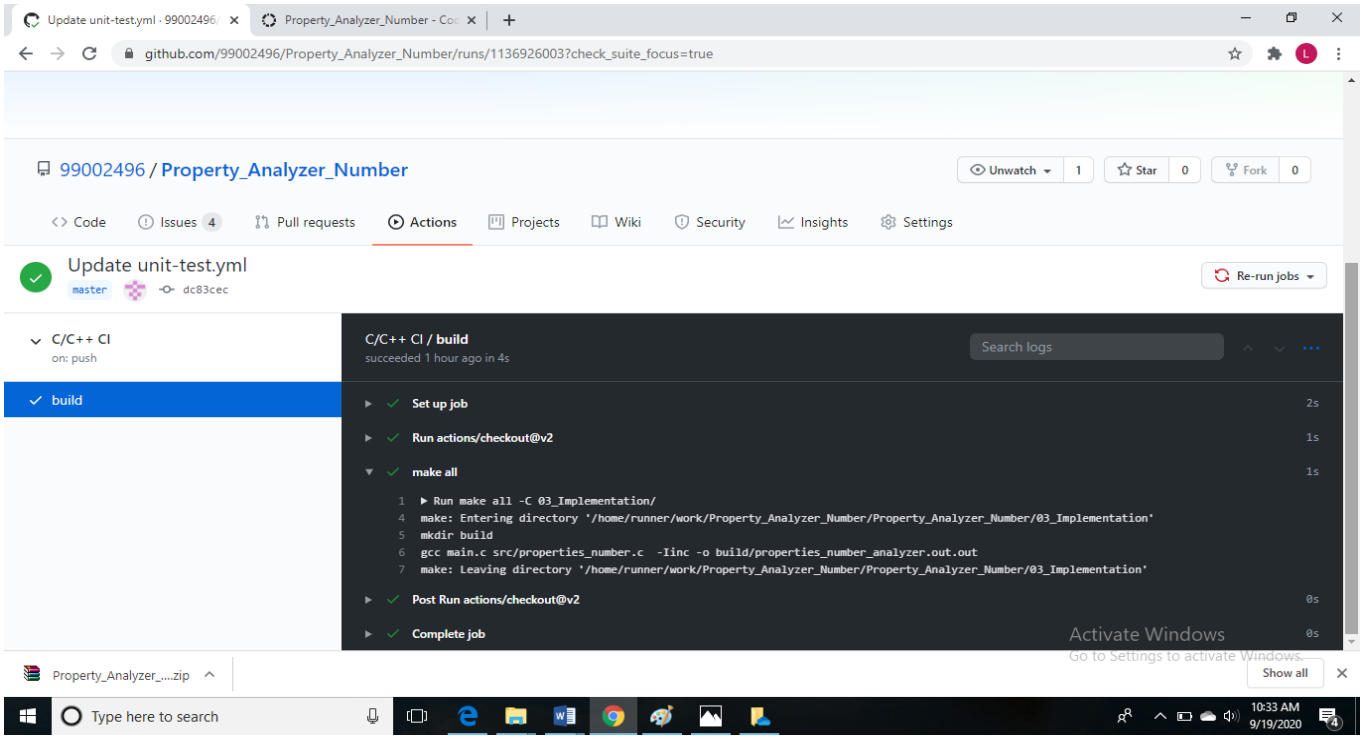
# CI WORKFLOWING

**Git Hub Link:** https://github.com/99002496/Property_Analyzer_Number.git

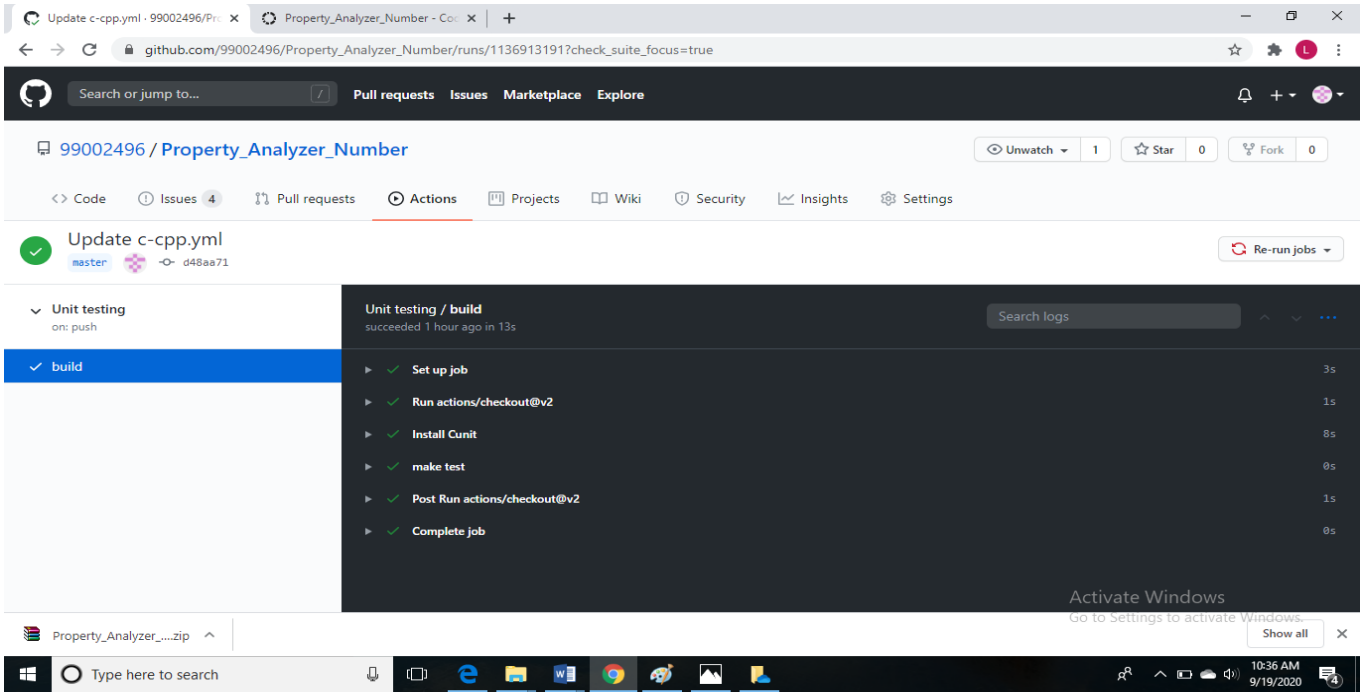
## CODE COMMITS



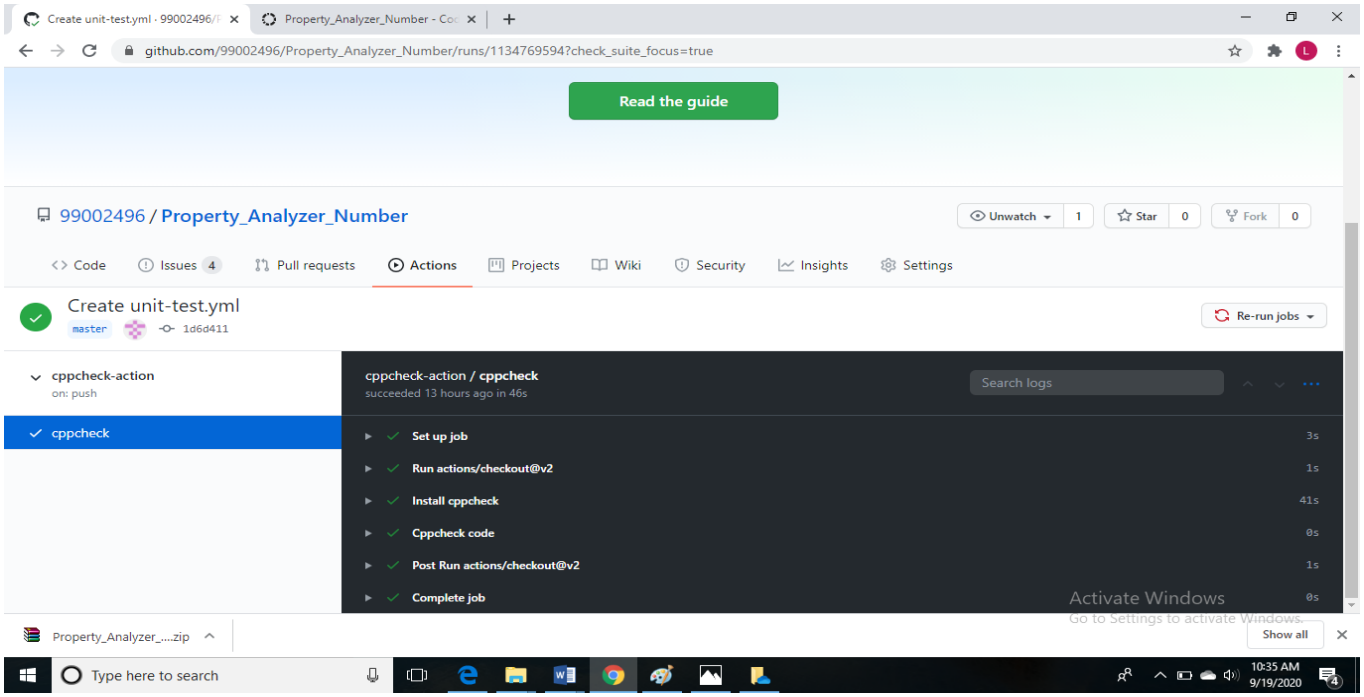
## BUILD



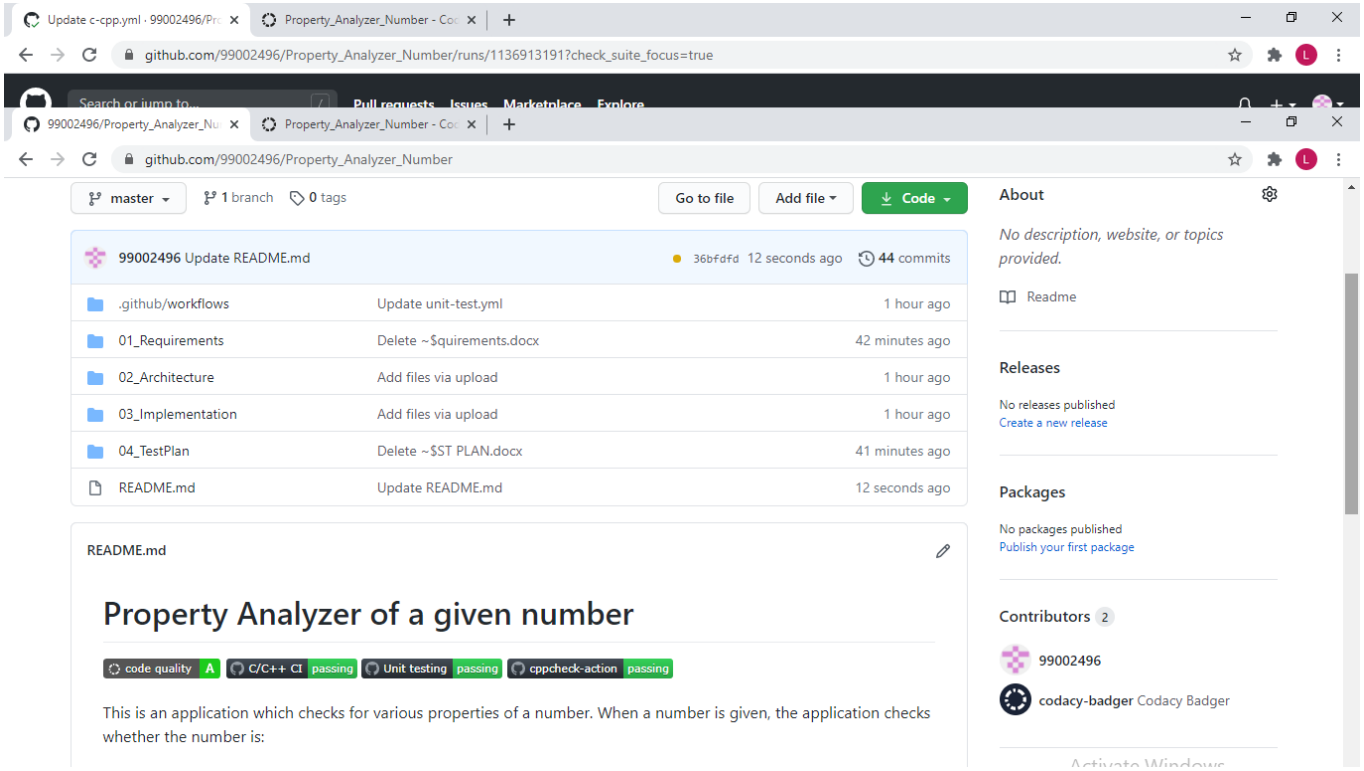

## UNIT TESTING



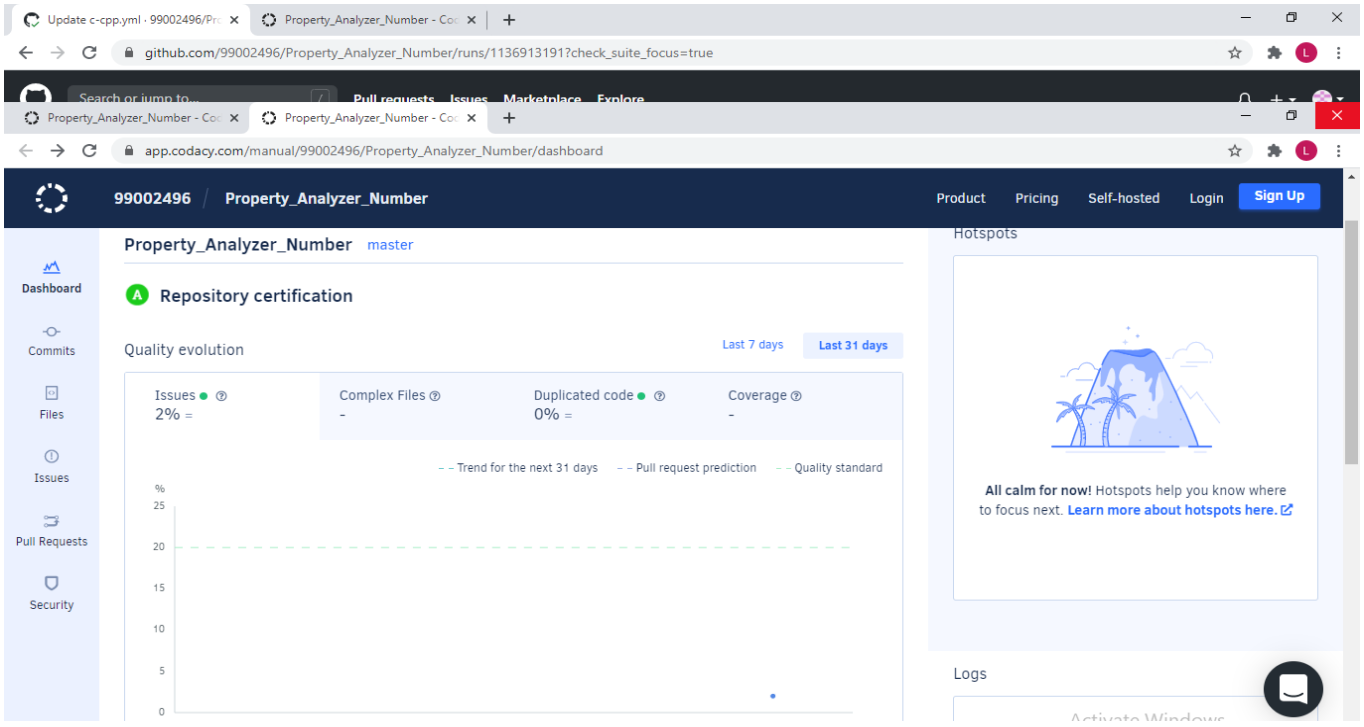
## CPPCHECK

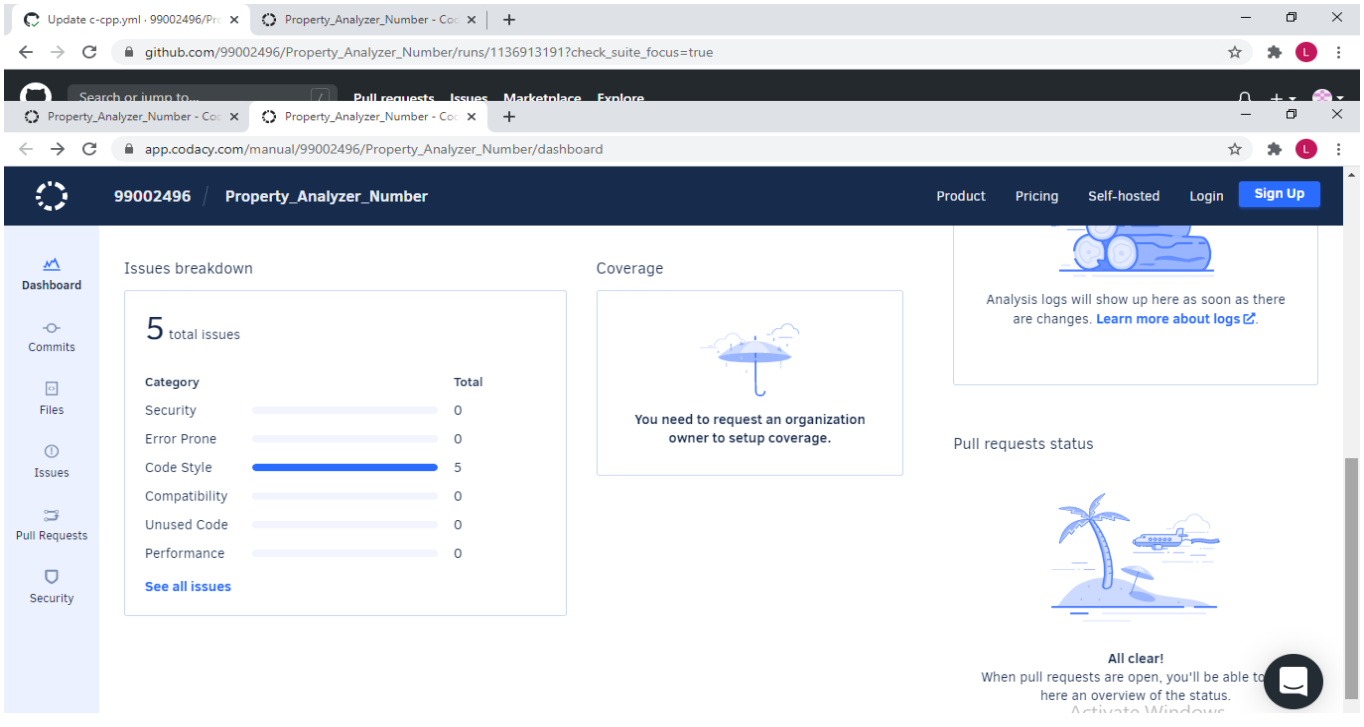



## BADGES



## CODACY


github.com/99002496/Property_Analyzer_Number/runs/1136913191?check_suite_focus=true
app.codacy.com/manual/99002496/Property_Analyzer_Number/dashboard
99002496 / Property_Analyzer_Number
Product
Pricing
Self-hosted
Login
Sign Up
Property_Analyzer_Number master
Repository certification
Quality evolution
Last 7 days
Last 31 days
Issues
2%
Complex Files
-
Duplicated code
0%
Coverage
-
Trend for the next 31 days
Pull request prediction
Quality standard
Hotspots
All calm for now! Hotspots help you know where to focus next. Learn more about hotspots here.
Logs
Dashboard
Commits
Files
Issues
Pull Requests
Security

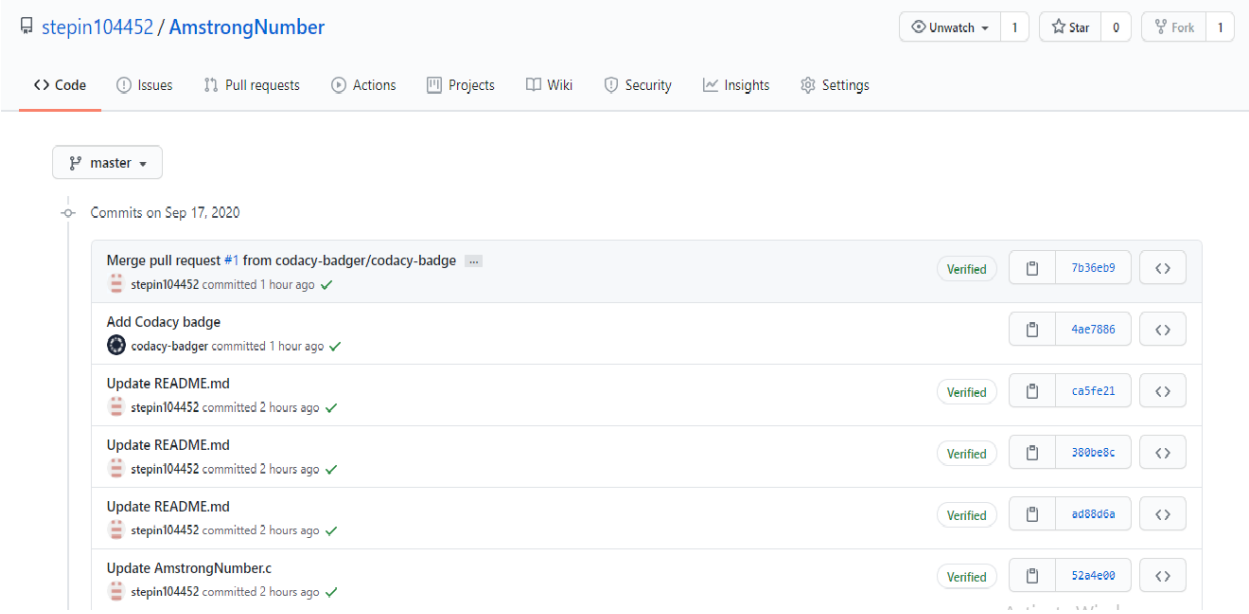

github.com/99002496/Property_Analyzer_Number/runs/1136913191?check_suite_focus=true
app.codacy.com/manual/99002496/Property_Analyzer_Number/dashboard
99002496 / Property_Analyzer_Number
Product
Pricing
Self-hosted
Login
Sign Up
Issues breakdown
5 total issues
Category
Total
Security 0
Error Prone 0
Code Style 5
Compatibility 0
Unused Code 0
Performance 0
See all issues
Coverage
You need to request an organization owner to setup coverage.
Analysis logs will show up here as soon as there are changes. Learn more about logs.
Pull requests status
All clear!
When pull requests are open, you'll be able to
here an overview of the status.
Dashboard
Commits
Files
Issues
Pull Requests
Security

# APPENDIX

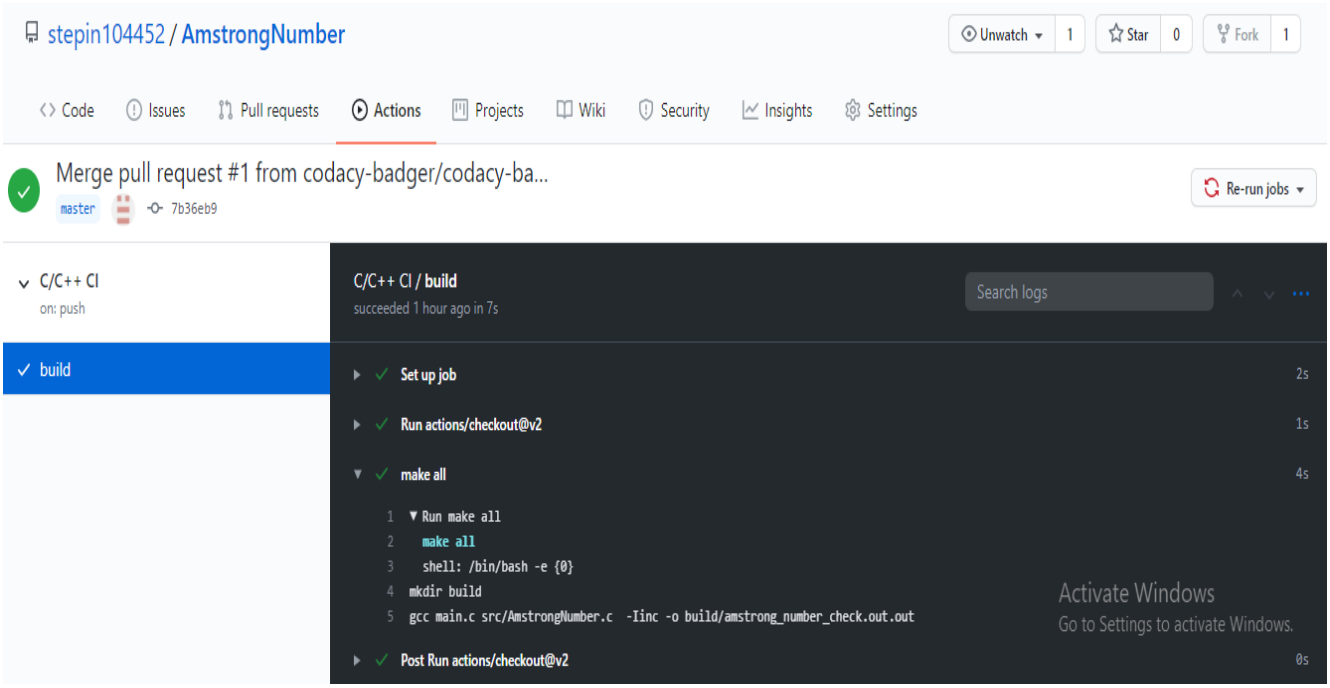
CI Workflow for C Programming

1) GIT HUB LINK: https://github.com/stepin104452/AmstrongNumber.git

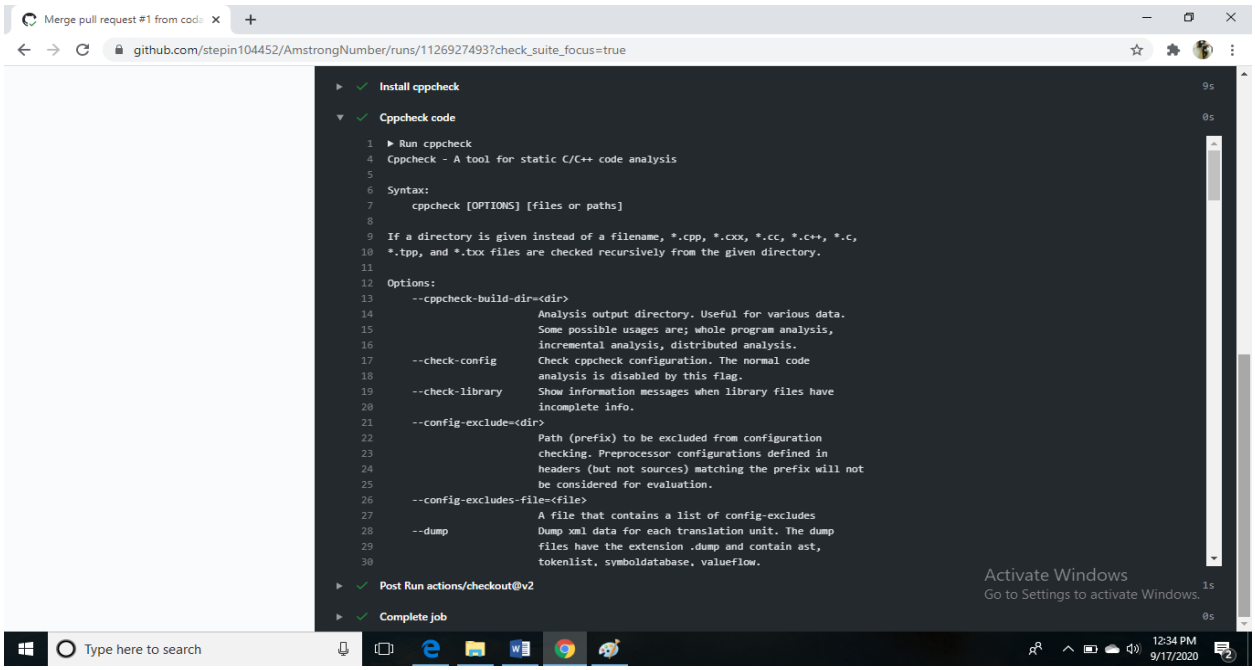
2) CODE COMMITS



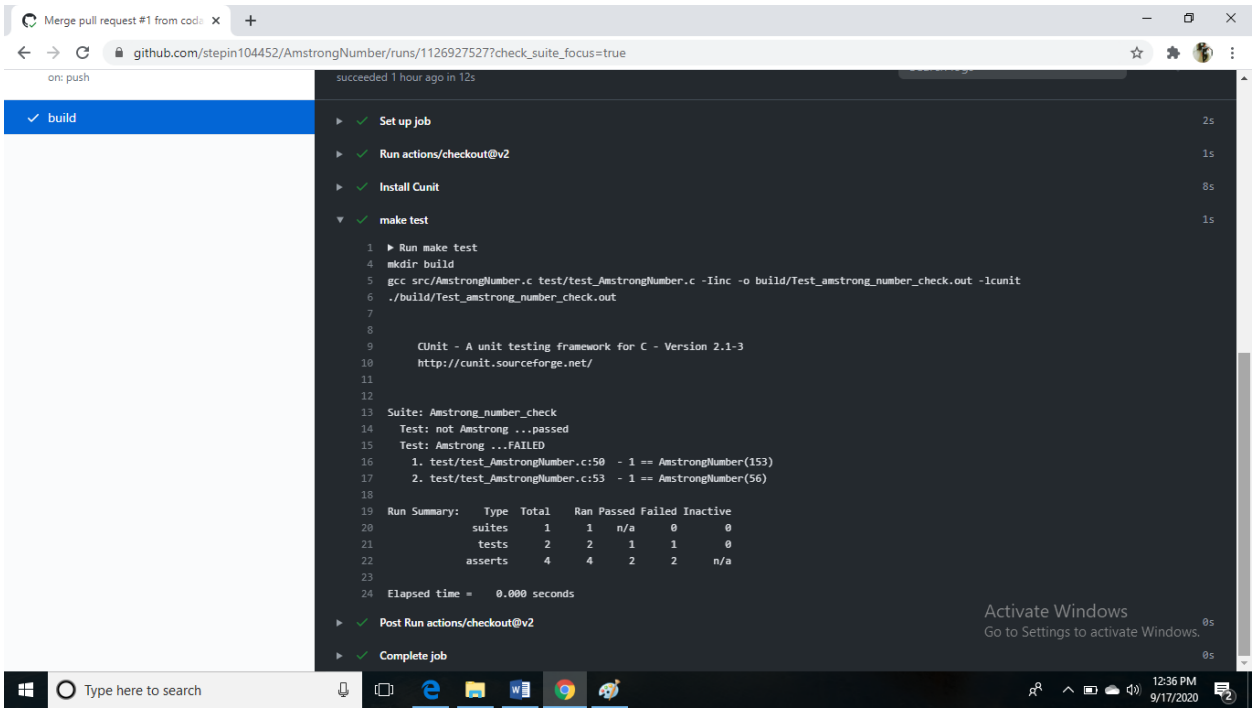
3) BUILD



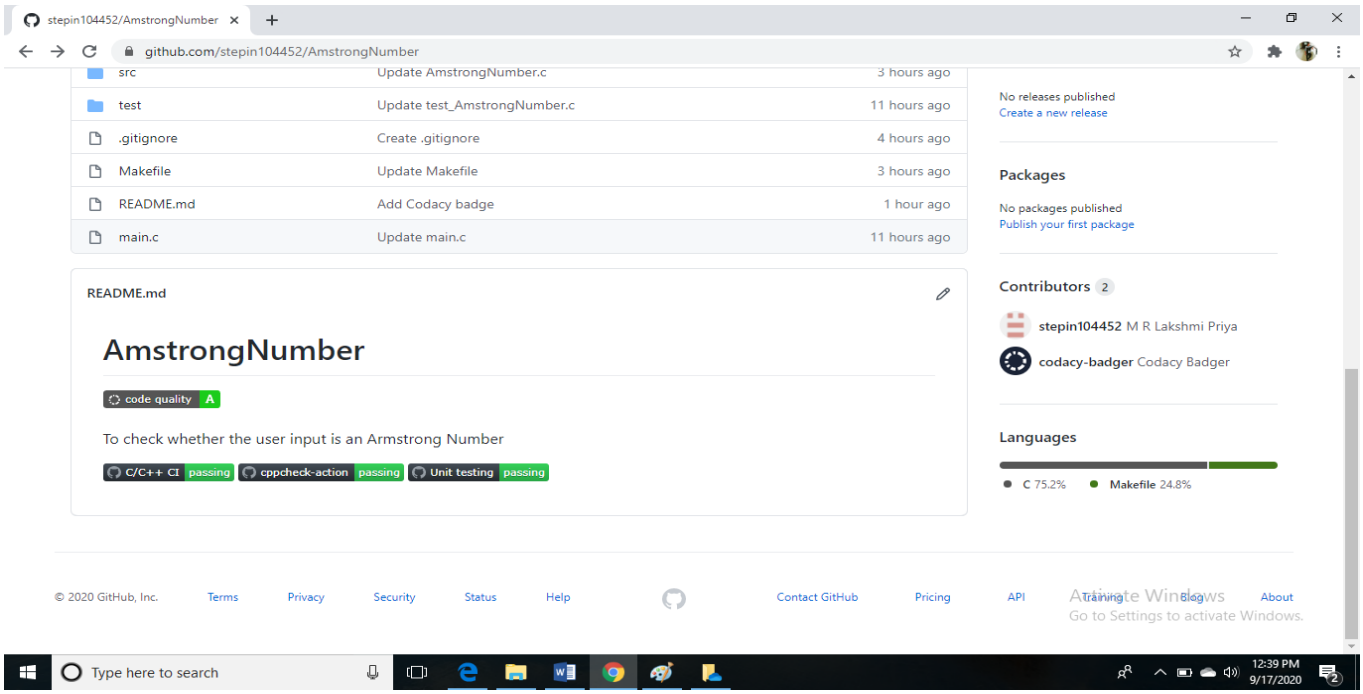

4) CPPCHECK



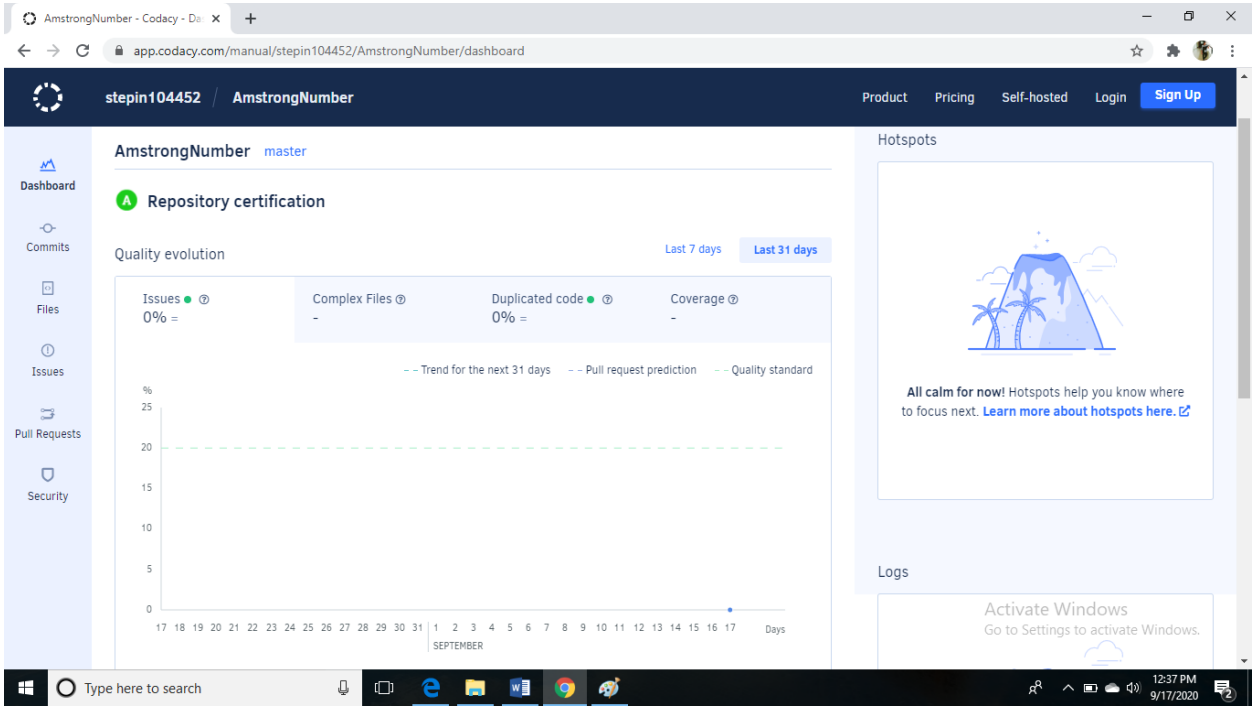
## 5) UNIT TESTING

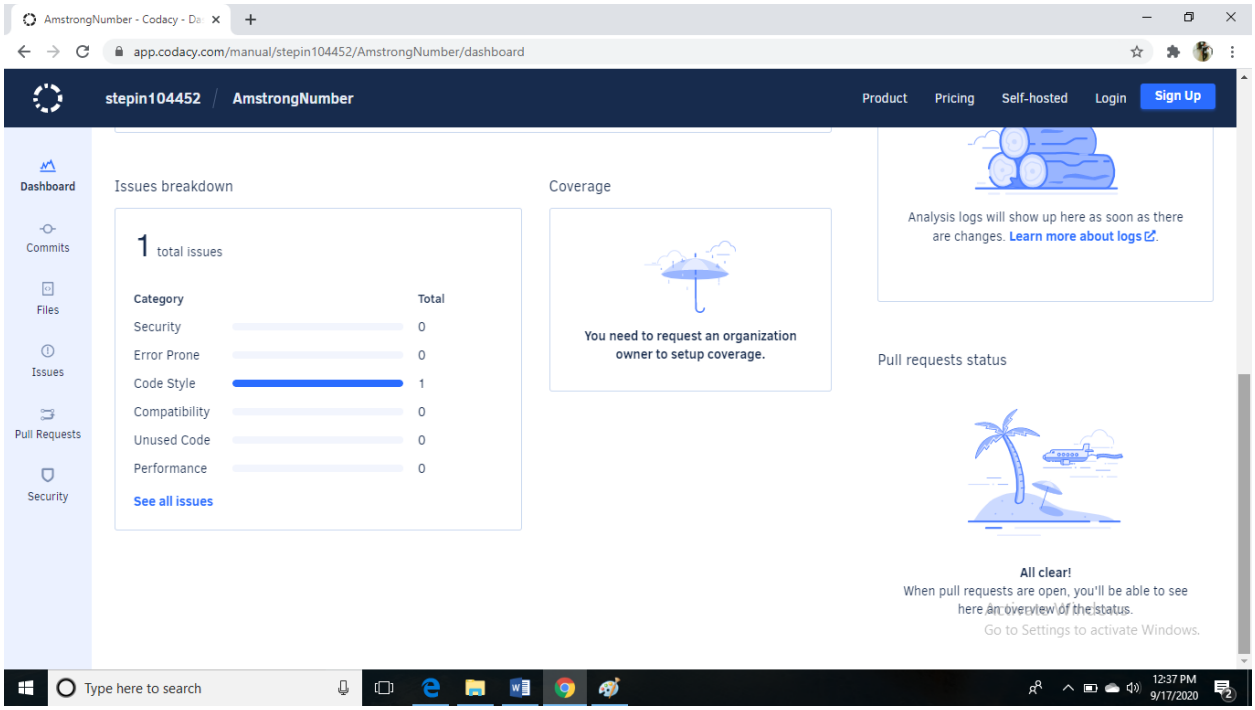



## 6) BADGES



## 7) CODACY


AmstrongNumber - Codacy - Da
app.codacy.com/manual/stepin104452/AmstrongNumber/dashboard
stepin104452 / AmstrongNumber
Product
Pricing
Self-hosted
Login
Sign Up
Dashboard
Commits
Files
Issues
Pull Requests
Security
AmstrongNumber master
Repository certification
Quality evolution
Last 7 days
Last 31 days
Issues
0%
Complex Files
-
Duplicated code
0%
Coverage
-
Trend for the next 31 days
Pull request prediction
Quality standard
Hotspots
All calm for now! Hotspots help you know where to focus next. Learn more about hotspots here.
Logs
Activate Windows
Go to Settings to activate Windows.
Type here to search
12:37 PM
9/17/2020


AmstrongNumber - Codacy - Da
app.codacy.com/manual/stepin104452/AmstrongNumber/dashboard
stepin104452 / AmstrongNumber
Product
Pricing
Self-hosted
Login
Sign Up
Dashboard
Commits
Files
Issues
Pull Requests
Security
Issues breakdown
1 total issues
Category
Total
Security 0
Error Prone 0
Code Style 1
Compatibility 0
Unused Code 0
Performance 0
See all issues
Coverage
You need to request an organization owner to setup coverage.
Analysis logs will show up here as soon as there are changes. Learn more about logs.
Pull requests status
All clear!
When pull requests are open, you'll be able to see here an overview of the status.
Go to Settings to activate Windows.
Type here to search
12:37 PM
9/17/2020

**MAIN CODE FILE**

```
#include <math.h>
#include<stdio.h>
#include "AmstrongNumber.h"

int main()
{
  int n, flag;
  printf("Enter a positive integer: ");
  scanf("%d", &n);
  flag = AmstrongNumber(n);
  if (flag == 1)
    printf("%d is an Amstrong number.", n);
  else
    printf("%d is not an Amstrong number.", n);
  return 0;
}
```

**FUNCTION CODE FILE**

```
#include <math.h>
#include "AmstrongNumber.h"

int ArmstrongNumber(int num)
{
  int originalNum, remainder, n = 0, flag;
  double result = 0.0;
  for (originalNum = num; originalNum != 0; ++n)
       originalNum /= 10;

  for (originalNum = num; originalNum != 0; originalNum /= 10)
{
    remainder = originalNum % 10;
    result += pow(remainder, n);
  }
  if (round(result) == num)
    flag = 1;
  else
    flag = 0;
  return flag;
}
```

**HEADER CODE**

```
#ifndef __AMSTRONGNUMBER_H__
#define __AMSTRONGNUMBER_H__

int AmstrongNumber(int number);

#endif /* #ifndef __AMSTRONGNUMBER_H__ */
```

**MAKE FILE**

```
TEST_SRC = src/AmstrongNumber.c\
test/test_AmstrongNumber.c
TEST_OUTPUT = $(BUILD)/Test_$(PROJECT_NAME).out
INC       = -Iinc
PROJECT_OUTPUT = $(BUILD)/$(PROJECT_NAME).out
$(PROJECT_NAME):all
.PHONY: run clean test  doc all

all: $(SRC) $(BUILD)
        gcc $(SRC) $(INC) -o $(PROJECT_OUTPUT).out

run:$(PROJECT_NAME)
        ./$(PROJECT_OUTPUT).out

test:$(BUILD)
        gcc $(TEST_SRC) $(INC) -o $(TEST_OUTPUT) -lcunit
        ./$(TEST_OUTPUT)

clean:
        rm -rf $(BUILD)

$(BUILD):
        mkdir build
```