



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

TASK 1- SMART LOCK FOR YOUR SECURITY AND SAFETY	4
1. RESEARCH	4
1.1 Ageing	4
1.2 Costing	4
2. DEFINITION OF PRODUCT	4
3. SWOT ANALYSIS	5
4. REQUIREMENTS	6
5. SYSTEM DESIGN	7
5.1 Structural Diagram.....	7
5.2 Behavioral Diagram	9
6. MODULAR DESIGN	11
6.1 Structural Diagram.....	11
6.2 Behavioral Diagram	13
7. TEST PLAN	15
7.1 Integration Test Plan and Unit Test Plan.....	15
7.2 Requirement Based	17
7.3 Scenario Based	17
7.4 Boundary Condition Based.....	17
8. REFERENCES.....	17
TASK 2- AGILE MODEL	18
1. THEME.....	18
2. EPIC	18
3. USER STORIES.....	18
4. REFERENCES.....	19
TASK 3- MINIPROJECT.....	20
1. INTRODUCTION.....	20
2. REQUIREMENTS	20
2.1 High Level Requirements.....	20
2.2 Low Level Requirements	21
3. UML DIAGRAM.....	22
4. TEST PLAN.....	26
5. CI WORKFLOWING	29
5.1 Git Hub Link	29
5.2 Code Commits	29
5.3 Build	30
5.4 CppCheck.....	30
5.5 Unit Testing.....	31
5.6 Badges.....	31
5.7 Codacy	32
APPENDIX	33

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

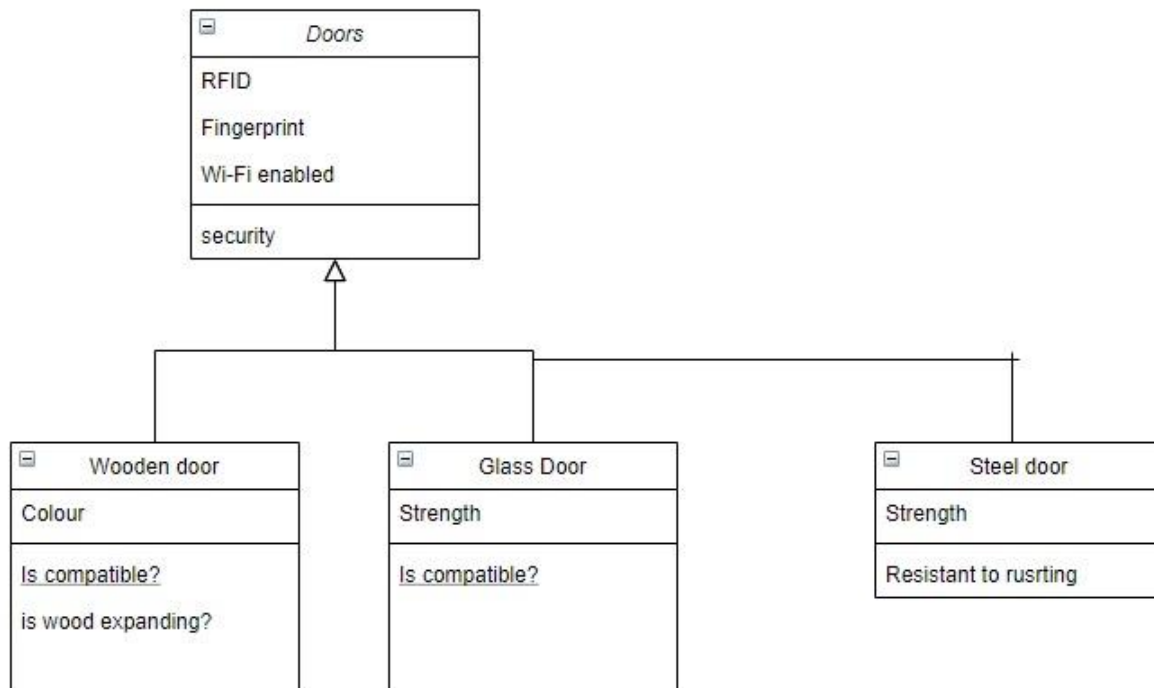
REQUIREMENTS

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

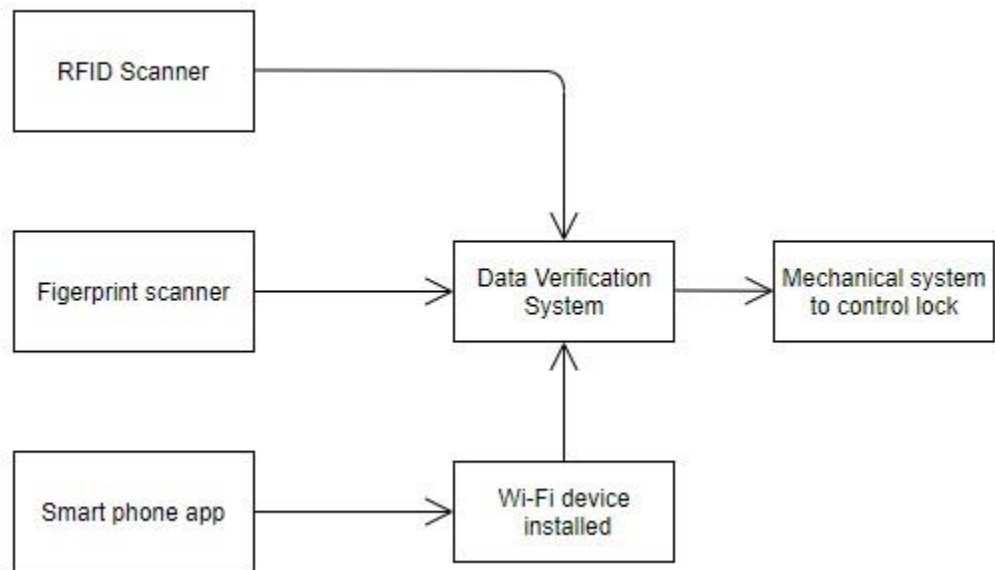
SYSTEM DESIGN

1) Structural Diagram

- CLASS DIAGRAM

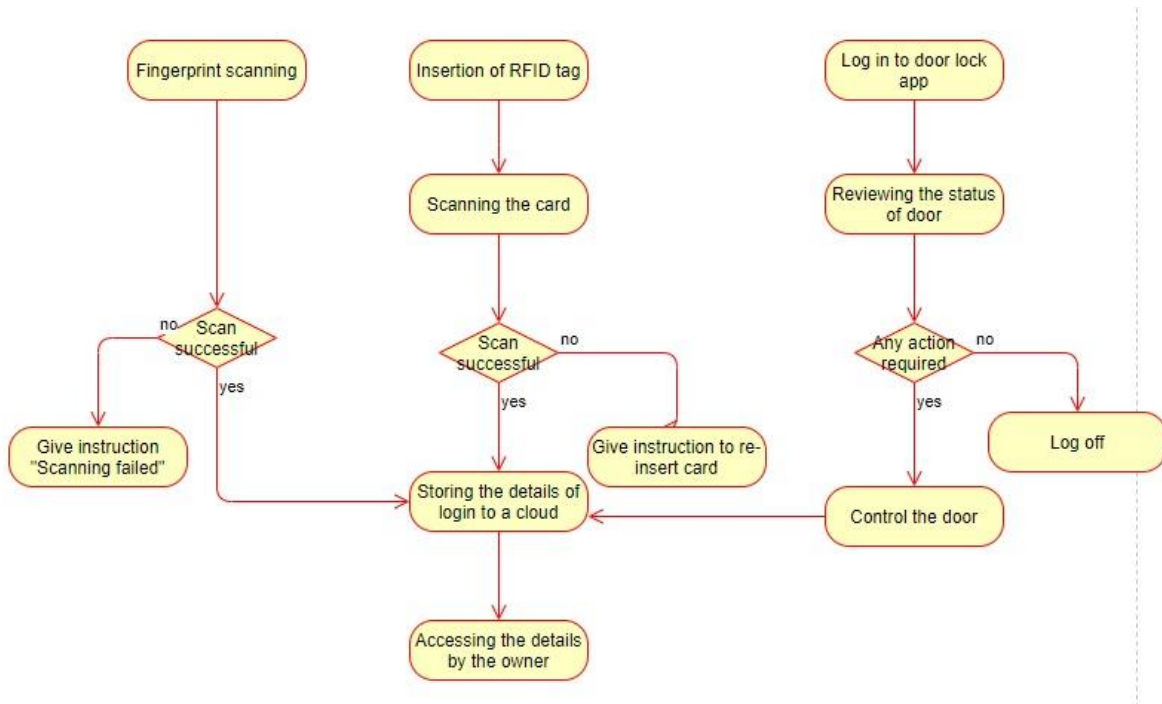


- COMPONENT DIAGRAM

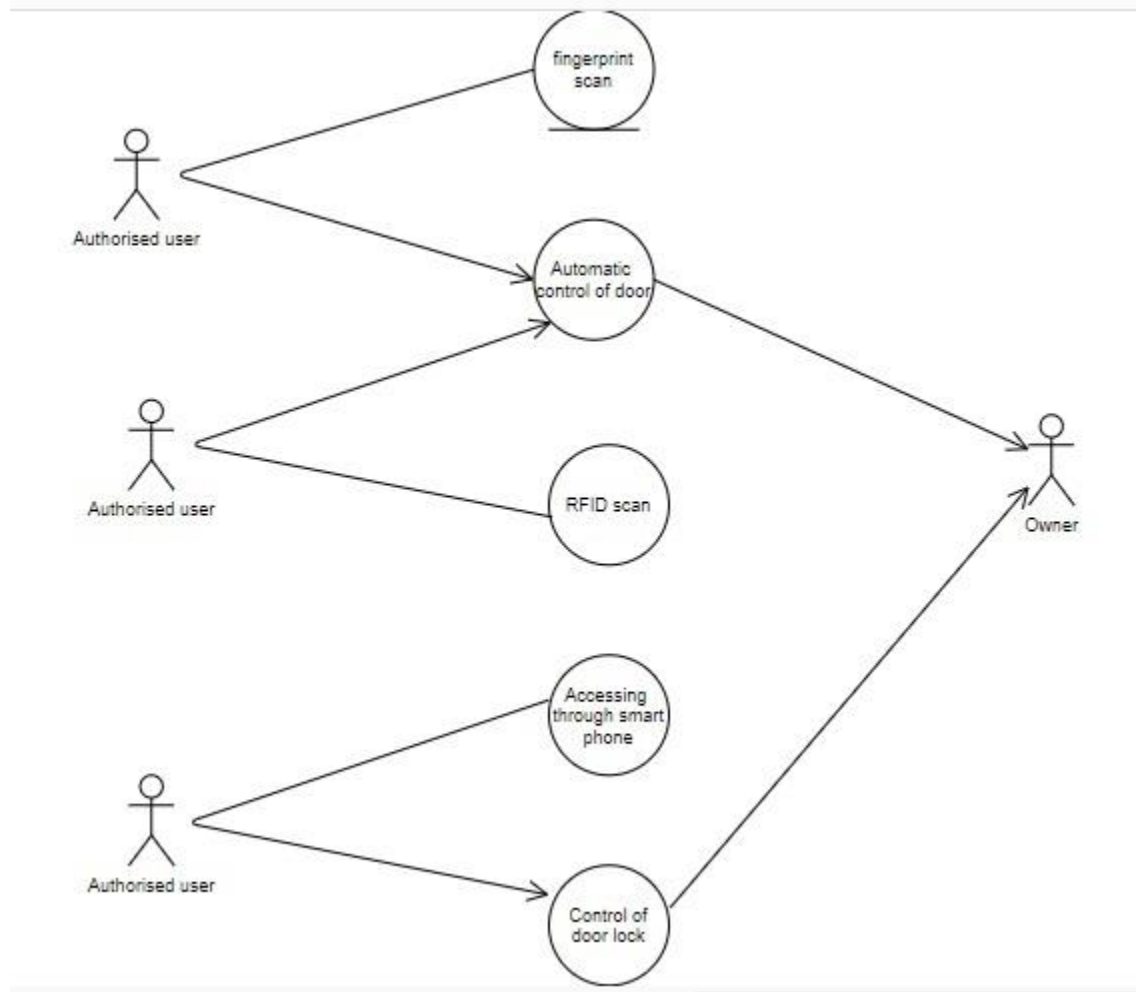


2) Behavioral Diagram

- ACTIVITY DIAGRAM

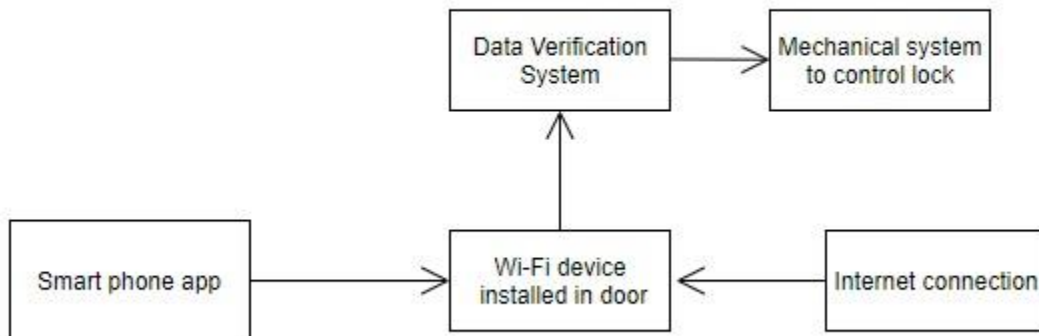


- USECASE DIAGRAM

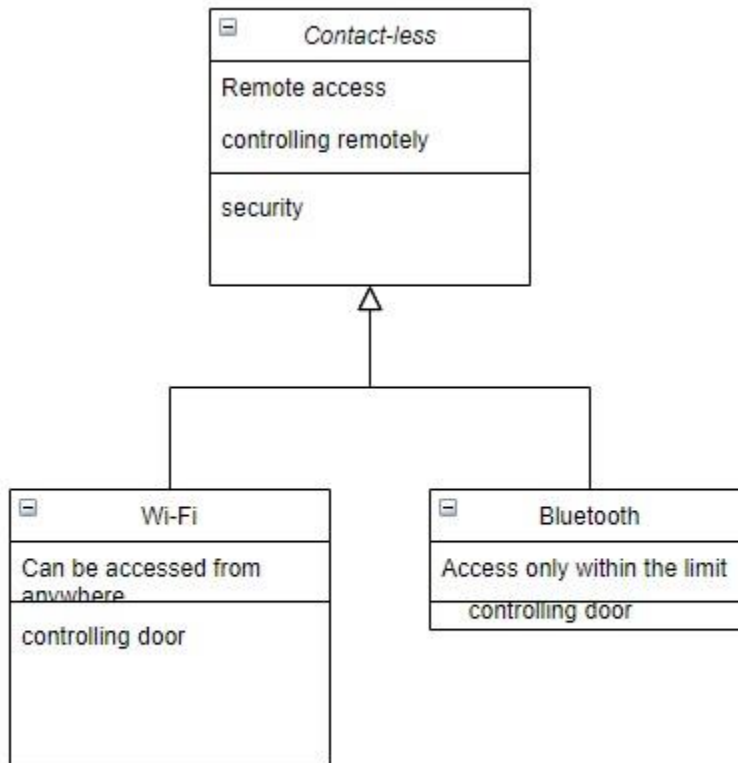


MODULAR DESIGN**1) Structural Diagram**

- COMPONENT DIAGRAM

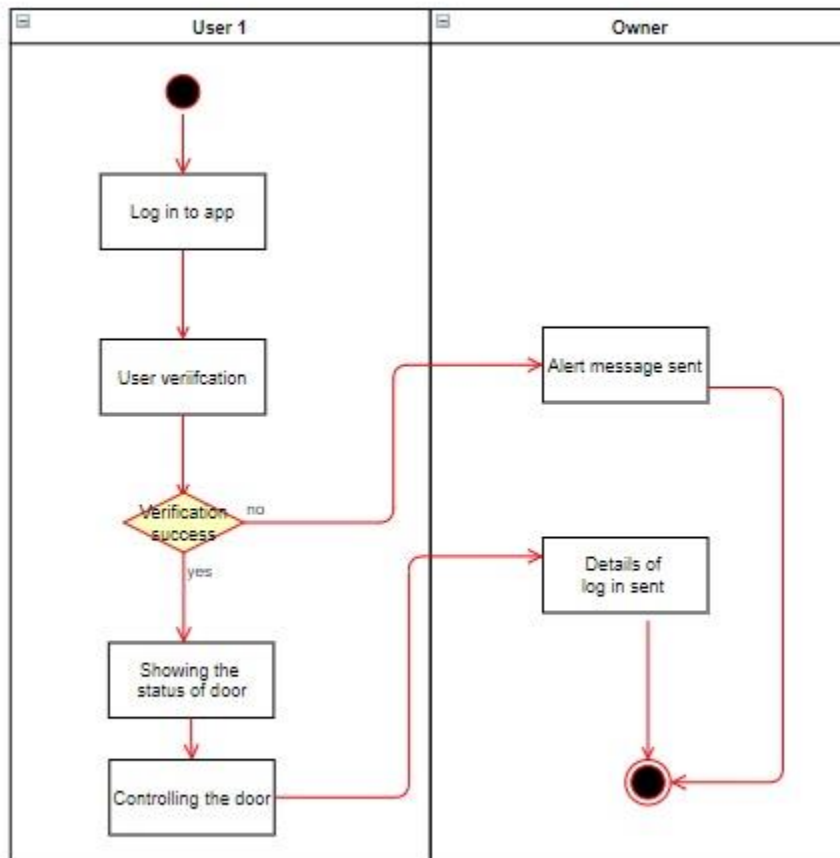


- CLASS DIAGRAM

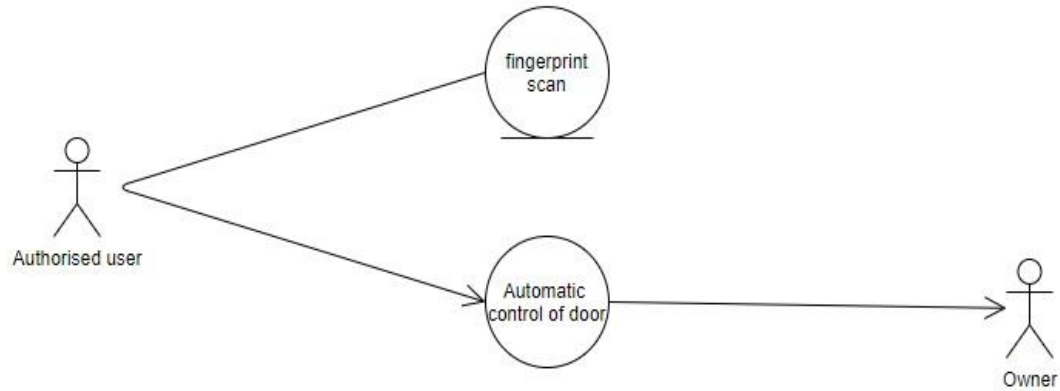


2) Behavioral Diagram

- 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-buying-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 hence 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**

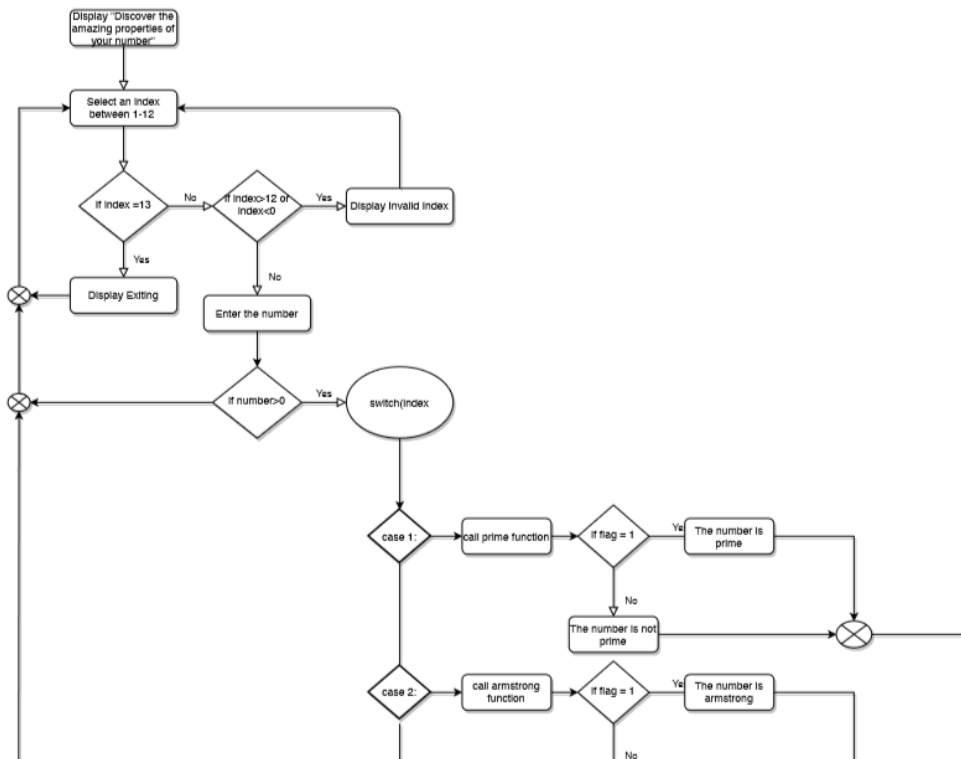
This is an application which checks for various properties of a number. When a number is given, the application checks whether the number is Prime, Armstrong, Odd/Even, Palindrome, Power, Harshad, Perfect_square, Perfect_cube, Automorphic, Divisible by 3, Divisible by 5 and Divisible by 7.

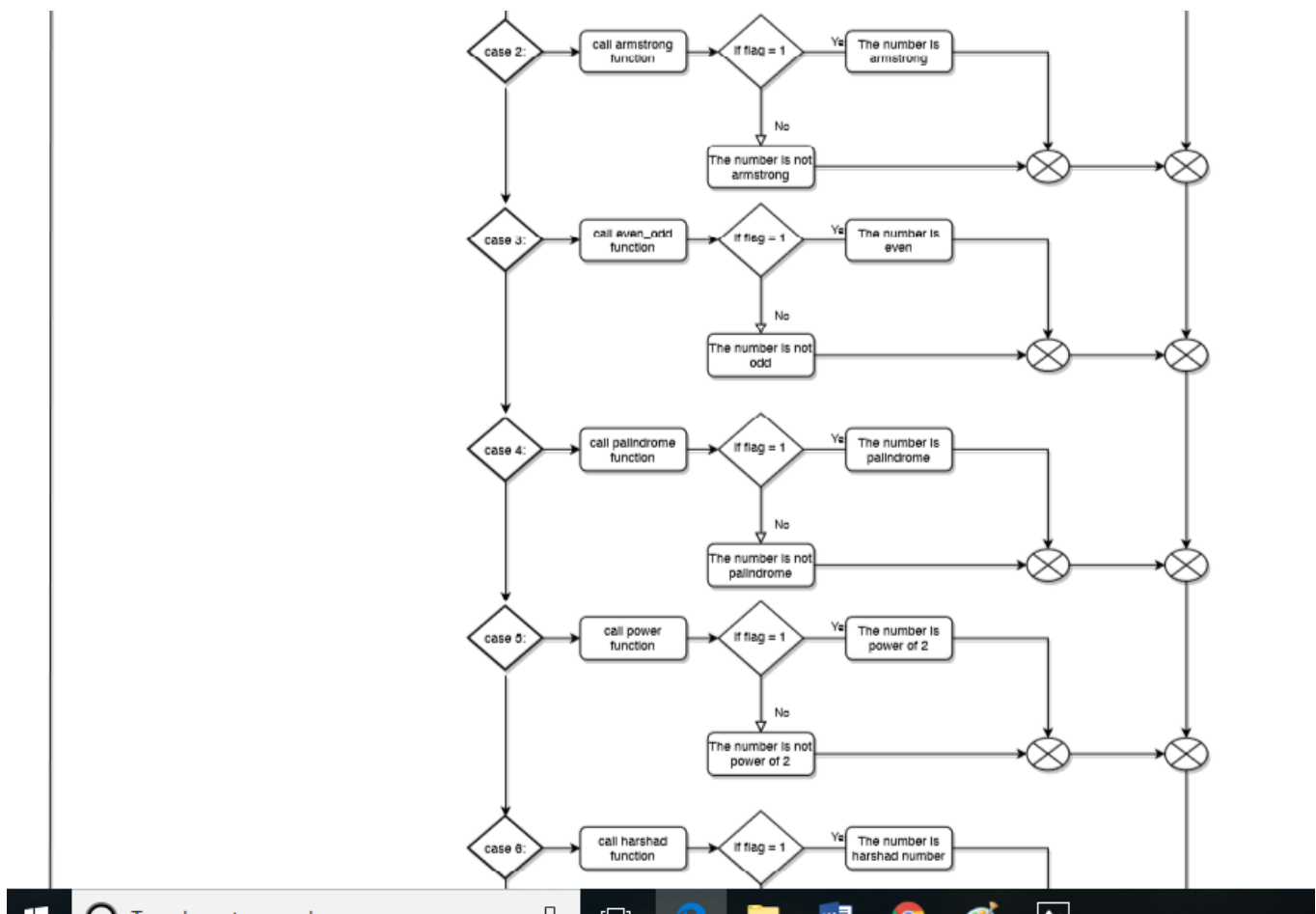
REQUIREMENTS**HIGH LEVEL REQUIREMENTS**

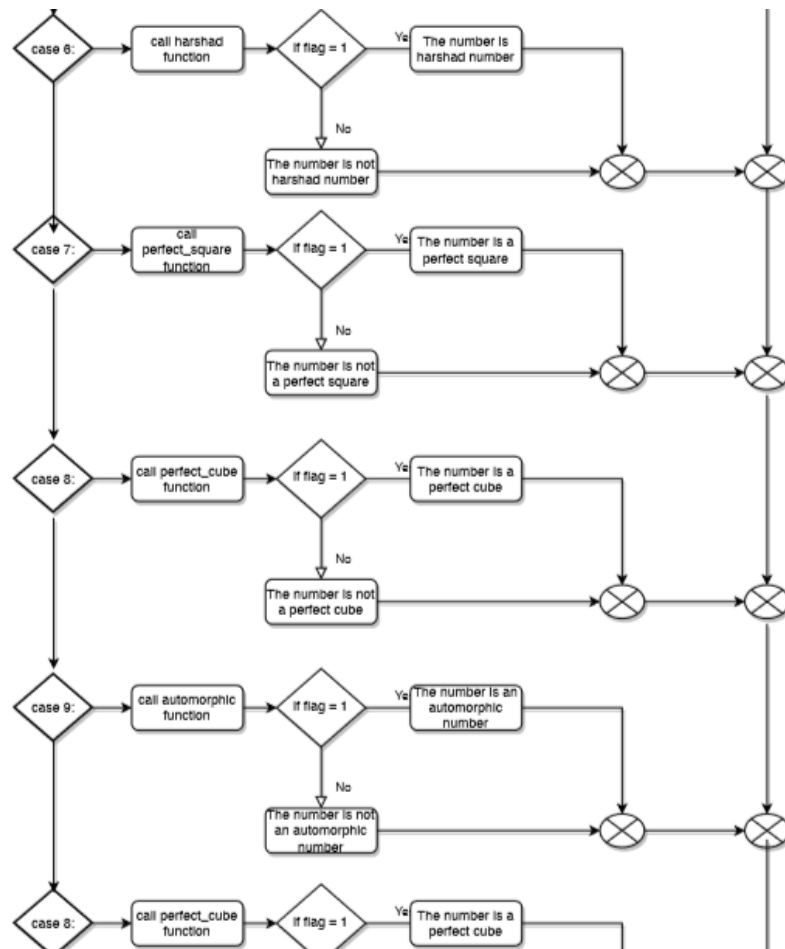
ID	DESCRIPTION
HL_01	Check for different properties of the number
HL_02	Determine the divisibility
HL_03	Perform different operations

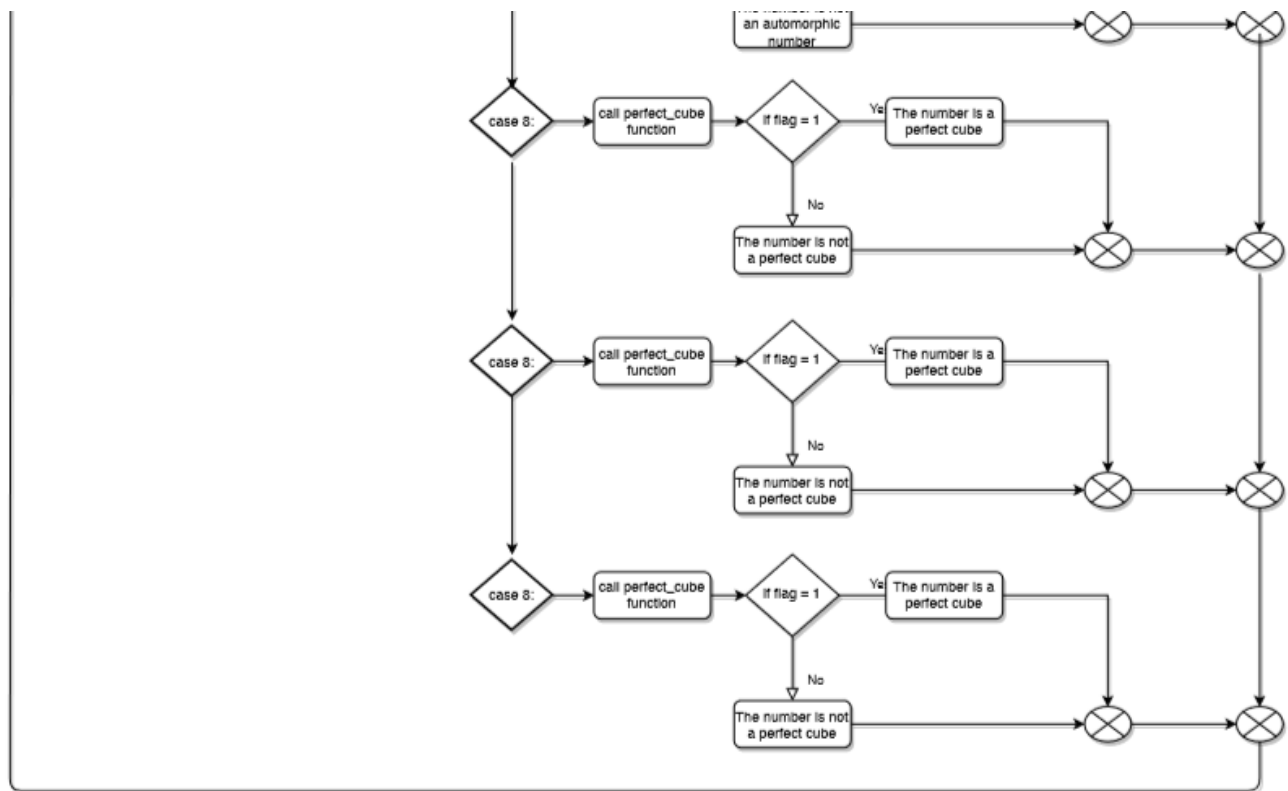
LOW LEVEL REQUIREMENTS

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 DIAGRAMAct
Go t







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	

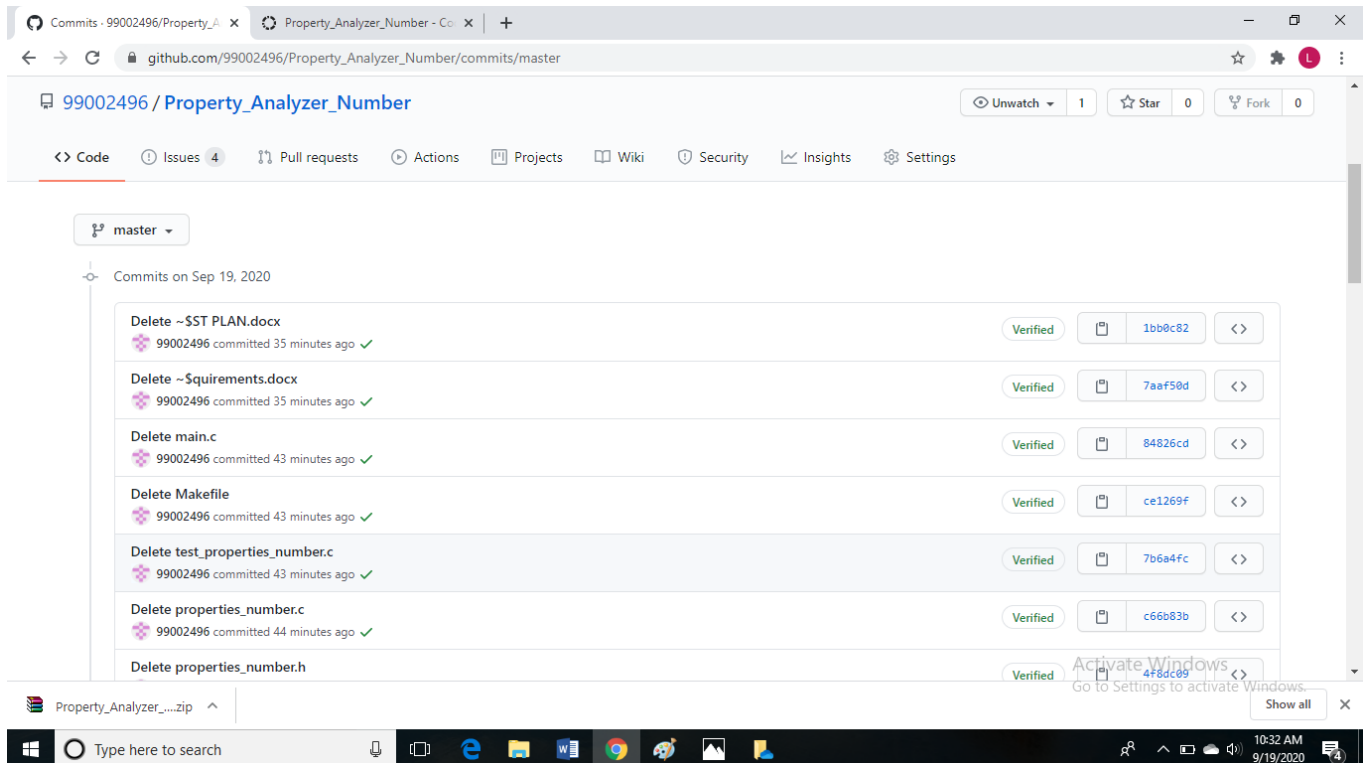
HL_01_LL_04_UT_02	Testing with a number which is not a palindrome	The number must be positive	15	The number is not Palindrome	
HL_01_LL_05_UT_01	Testing with a number which is a power of 2	The number must be positive	4	The number is a power of 2	
HL_01_LL_05_UT_02	Testing with a number which is not a 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 12	27	
HL_03_LL_02_UT_01	Testing Subtraction	The number must be positive	48 5	43	
HL_03_LL_03_UT_01	Testing Multiplication	The number must be positive	5 6	30	
HL_03_LL_04_UT_01	Testing Division	The number must be positive	25 5	5	

CI WORKFLOWING

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

CODE COMMITS



Commits · 99002496/Property_Analyzer_Number

99002496 / Property_Analyzer_Number

Unwatch 1 Star 0 Fork 0

Code Issues 4 Pull requests Actions Projects Wiki Security Insights Settings

master

Commits on Sep 19, 2020

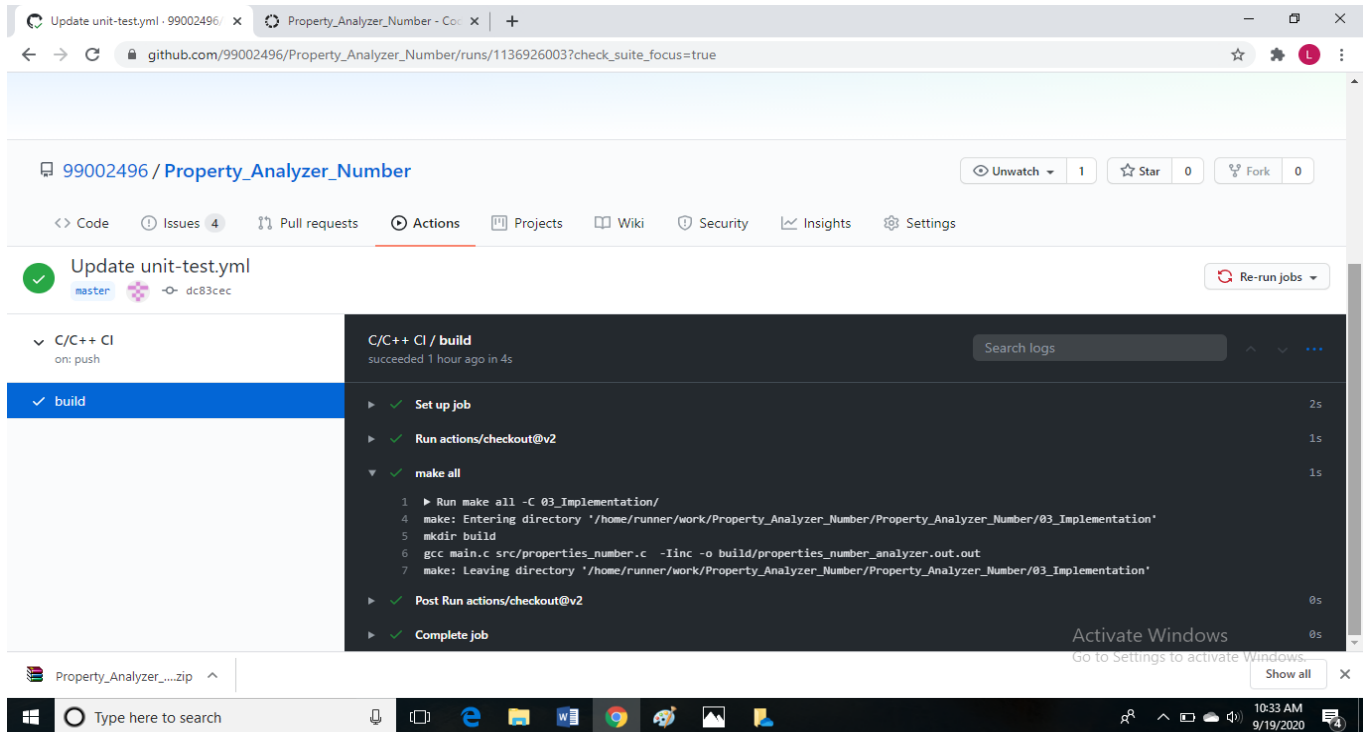
Delete ~\$ST PLAN.docx 99002496 committed 35 minutes ago ✓	Verified	1bb0c82	<>
Delete ~\$quirements.docx 99002496 committed 35 minutes ago ✓	Verified	7aaf50d	<>
Delete main.c 99002496 committed 43 minutes ago ✓	Verified	84826cd	<>
Delete Makefile 99002496 committed 43 minutes ago ✓	Verified	ce1269f	<>
Delete test_properties_number.c 99002496 committed 43 minutes ago ✓	Verified	7b6a4fc	<>
Delete properties_number.c 99002496 committed 44 minutes ago ✓	Verified	c66b83b	<>
Delete properties_number.h 99002496 committed 44 minutes ago ✓	Verified	4f8dc09	<>

Property_Analyzer_...zip

Type here to search

10:32 AM 9/19/2020

BUILD



Update unit-test.yml · 99002496 / Property_Analyzer_Number · Code · +

github.com/99002496/Property_Analyzer_Number/runs/1136926003?check_suite_focus=true

99002496 / Property_Analyzer_Number

Unwatch 1 Star 0 Fork 0

Code Issues 4 Pull requests Actions Projects Wiki Security Insights Settings

Update unit-test.yml
master dc83cec

Re-run jobs

C/C++ CI / build
on: push

build

C/C++ CI / build
succeeded 1 hour ago in 4s

Search logs

- Set up job 2s
- Run actions/checkout@v2 1s
- make all 1s
 - Run make all -C 03_Implementation/
 - make: Entering directory '/home/runner/work/Property_Analyzer_Number/Property_Analyzer_Number/03_Implementation'
 - mkdir build
 - gcc main.c src/properties_number.c -Iinc -o build/properties_number_analyzer.out
 - make: Leaving directory '/home/runner/work/Property_Analyzer_Number/Property_Analyzer_Number/03_Implementation'
- Post Run actions/checkout@v2 0s
- Complete job 0s

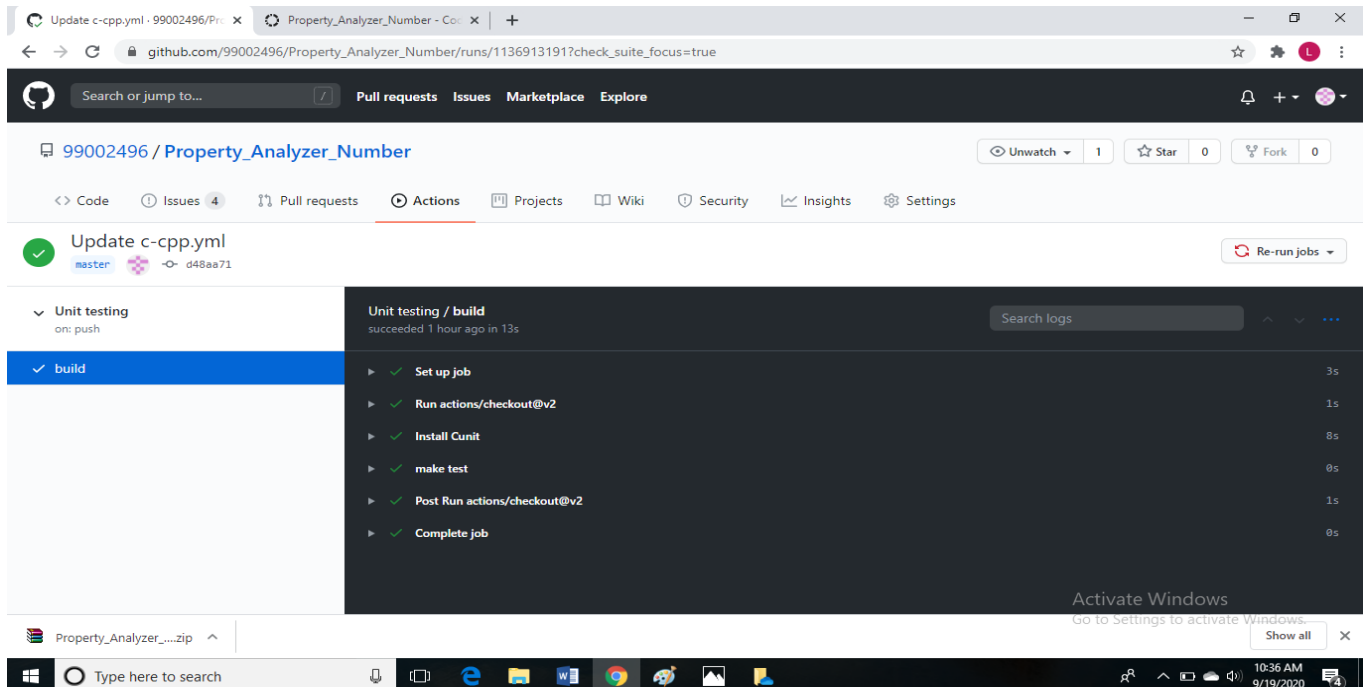
Activate Windows
Go to Settings to activate Windows.
Show all

Property_Analyzer_...zip

Type here to search

10:33 AM 9/19/2020

UNIT TESTING



Update c-cpp.yml · 99002496 / Property_Analyzer_Number · Code · +

github.com/99002496/Property_Analyzer_Number/runs/1136913191?check_suite_focus=true

99002496 / Property_Analyzer_Number

Unwatch 1 Star 0 Fork 0

Code Issues 4 Pull requests Actions Projects Wiki Security Insights Settings

Update c-cpp.yml
master d48aa71

Re-run jobs

Unit testing / build
on: push

build

Unit testing / build
succeeded 1 hour ago in 13s

Search logs

- Set up job 3s
- Run actions/checkout@v2 1s
- Install Cunit 8s
- make test 0s
- Post Run actions/checkout@v2 1s
- Complete job 0s

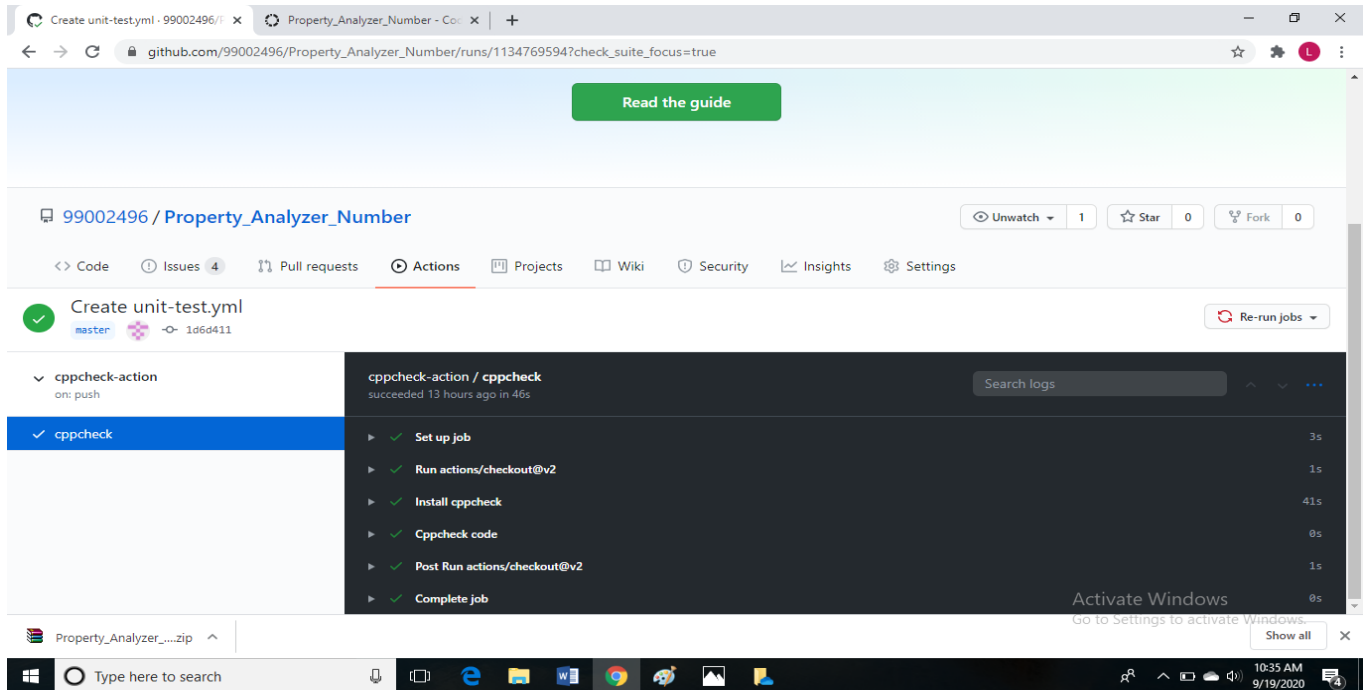
Activate Windows
Go to Settings to activate Windows.
Show all

Property_Analyzer_...zip

Type here to search

10:36 AM 9/19/2020

CPPCHECK

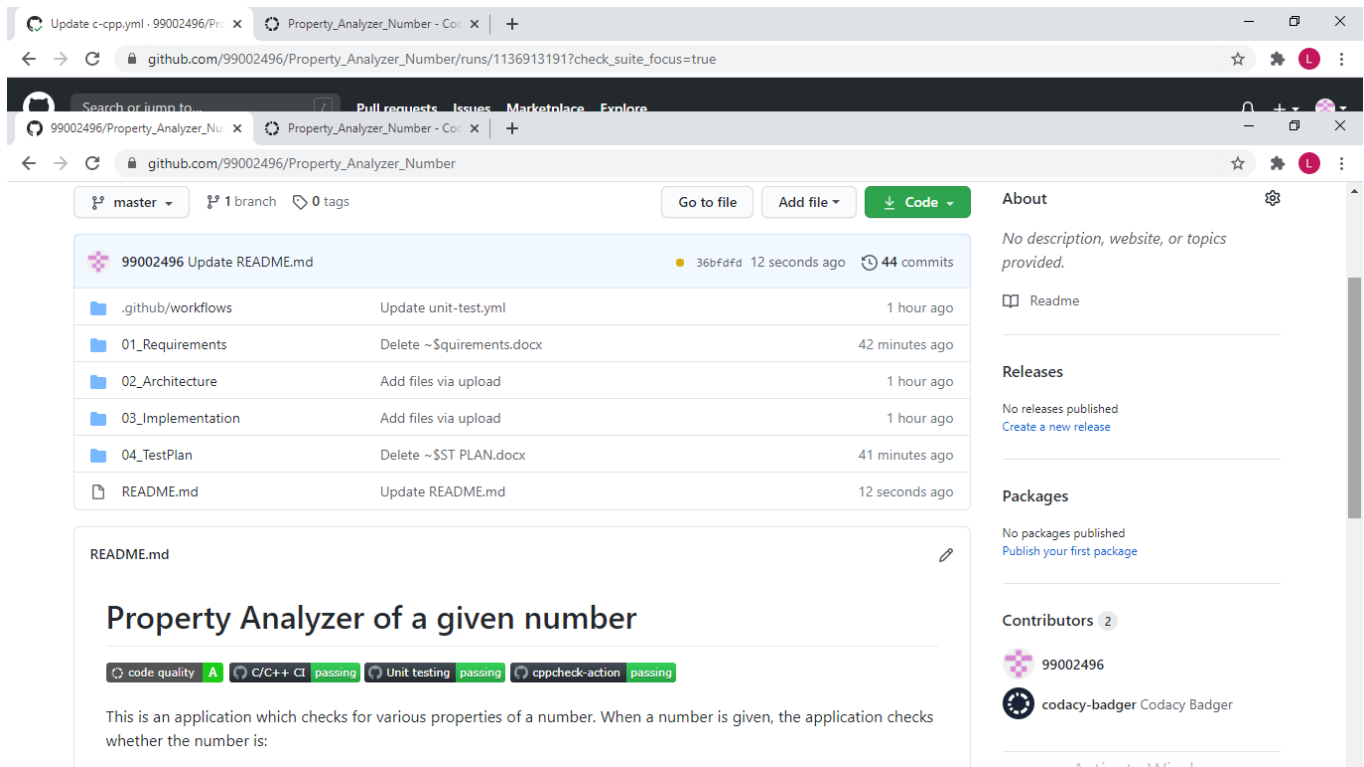


The screenshot shows a GitHub Actions workflow run for the repository 99002496/Property_Analyzer_Number. The workflow is named 'Create unit-test.yml' and is currently running on the master branch. The workflow steps are as follows:

- Set up job (3s)
- Run actions/checkout@v2 (1s)
- Install cppcheck (41s)
- Cppcheck code (0s)
- Post Run actions/checkout@v2 (1s)
- Complete job (0s)

The workflow succeeded 13 hours ago in 46s. The cppcheck-action step is highlighted in blue. The workflow is triggered on push.

BADGES

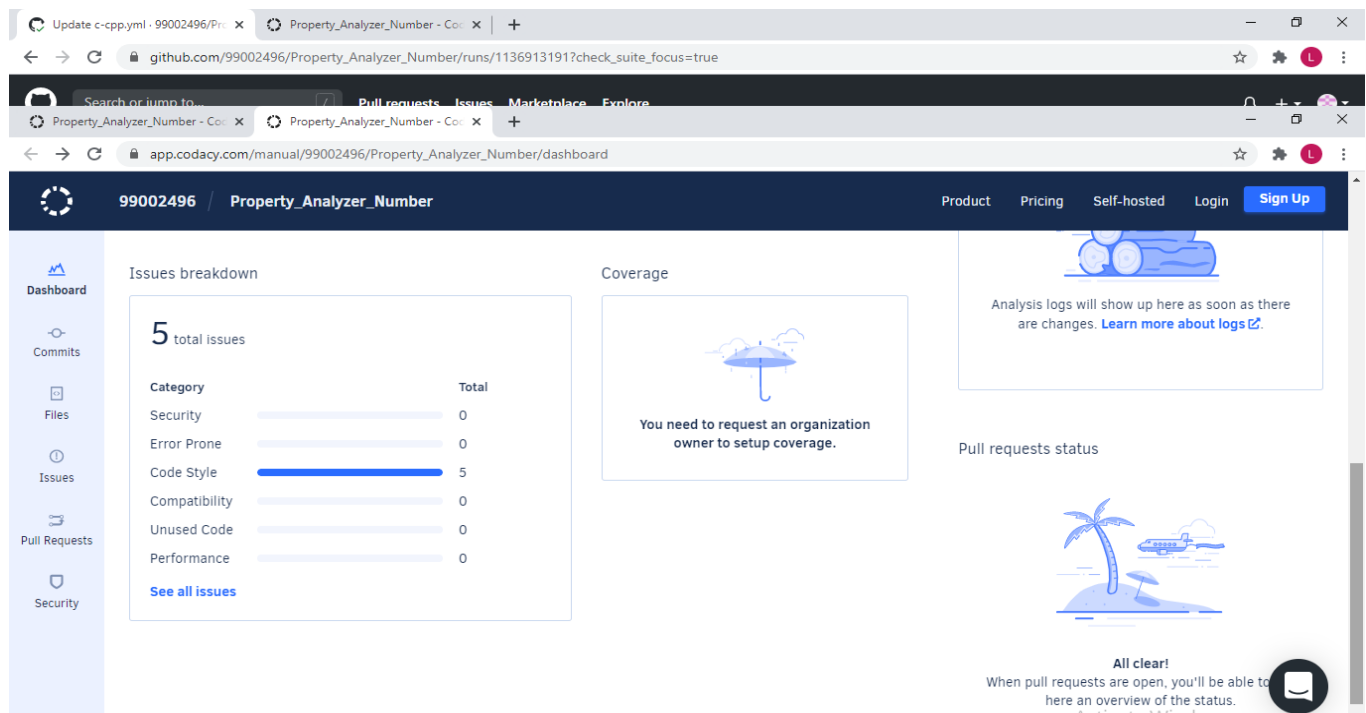
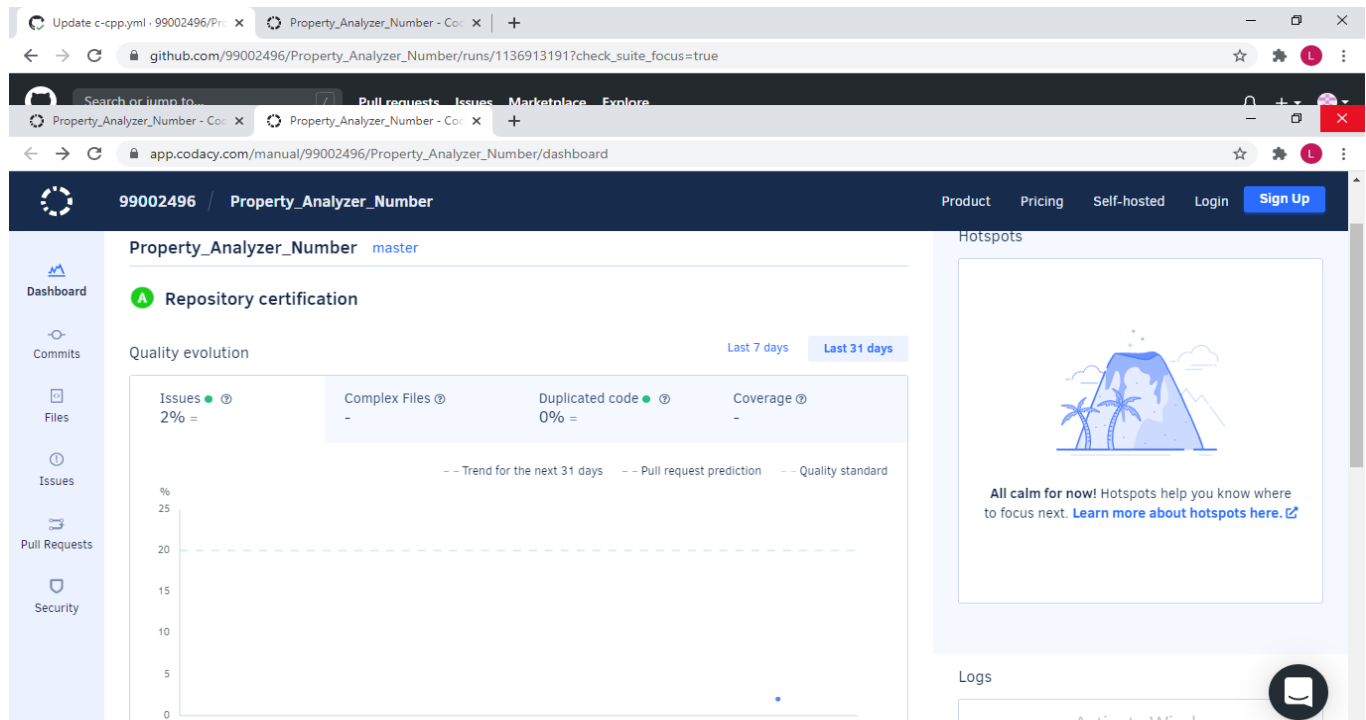


The screenshot shows the GitHub repository page for 99002496/Property_Analyzer_Number. The repository is currently on the master branch, which has 1 branch and 0 tags. The repository is a C++ project, and the workflow 'Create unit-test.yml' is running successfully. The repository contains the following files:

- .github/workflows
- 01_Requirements
- 02_Architecture
- 03_Implementation
- 04_TestPlan
- README.md

The README.md file is displayed, showing the title 'Property Analyzer of a given number' and a description: 'This is an application which checks for various properties of a number. When a number is given, the application checks whether the number is:'. The README also includes a badge for 'code quality' and a badge for 'C/C++ CI'.

CODACY

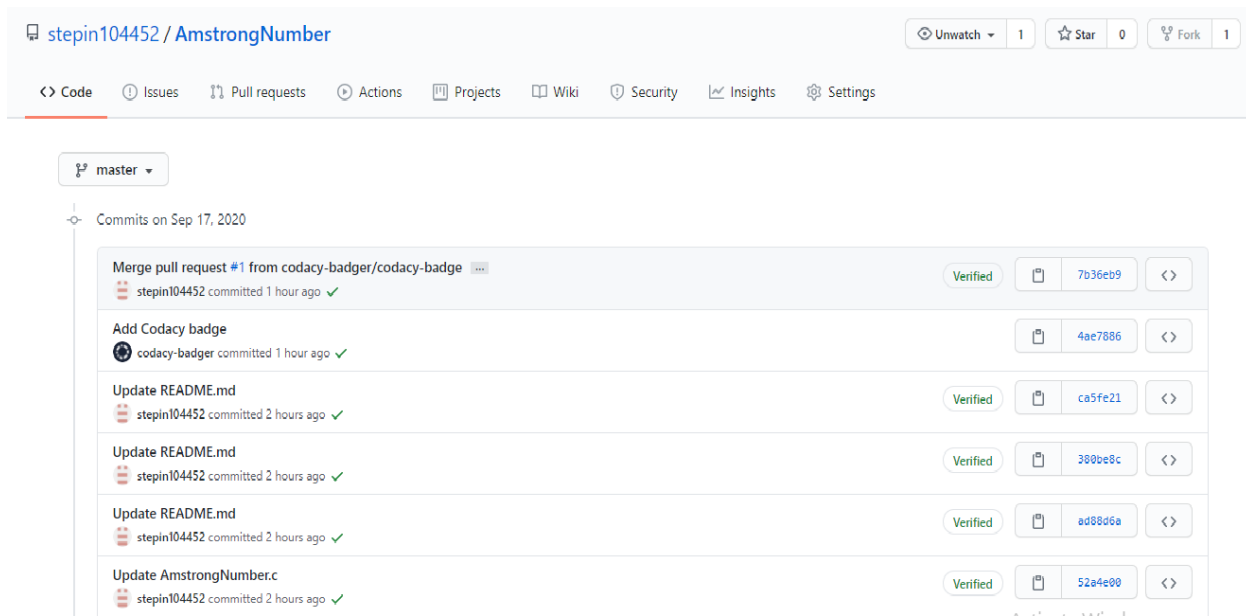


APPENDIX

CI Workflow for C Programming

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

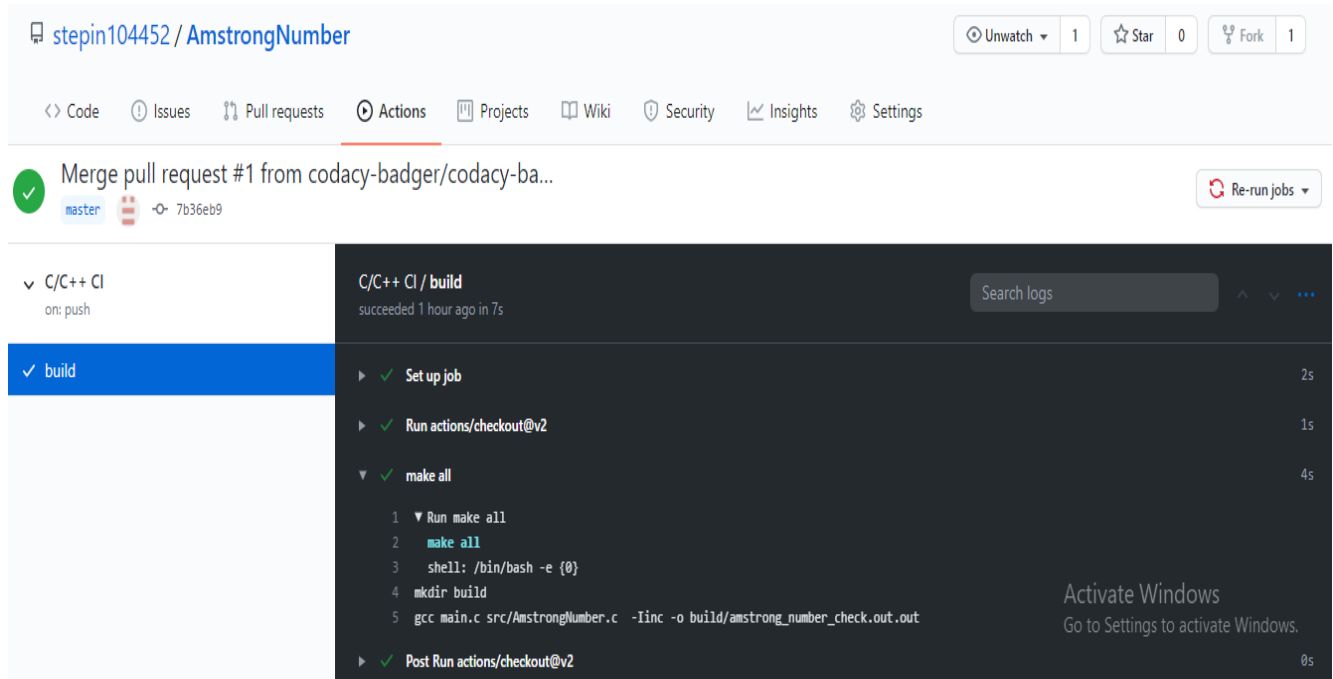
2) CODE COMMITS



The screenshot shows the GitHub repository page for `stepin104452 / AmstrongNumber`. The repository has 1 Unwatch, 1 Star, and 1 Fork. The navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The current branch is `master`. The commit history for September 17, 2020, is displayed, showing a series of commits by `stepin104452` and `codacy-badger`. Each commit includes a description, a timestamp, a verification status, and a commit hash.

Commit Description	Author	Timestamp	Verification	Commit Hash
Merge pull request #1 from codacy-badger/codacy-badger	stepin104452	committed 1 hour ago	Verified	7b36eb9
Add Codacy badge	codacy-badger	committed 1 hour ago		4ae7886
Update README.md	stepin104452	committed 2 hours ago	Verified	ca5fe21
Update README.md	stepin104452	committed 2 hours ago	Verified	380be8c
Update README.md	stepin104452	committed 2 hours ago	Verified	ad88d6a
Update AmstrongNumber.c	stepin104452	committed 2 hours ago	Verified	52a4e00

3) BUILD



stepin104452 / AmstrongNumber

Unwatch 1 Star 0 Fork 1

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Merge pull request #1 from codacy-badger/codacy-ba...
master 7b36eb9 Re-run jobs

C/C++ CI / build
succeeded 1 hour ago in 7s

Search logs

✓ build

Set up job 2s

Run actions/checkout@v2 1s

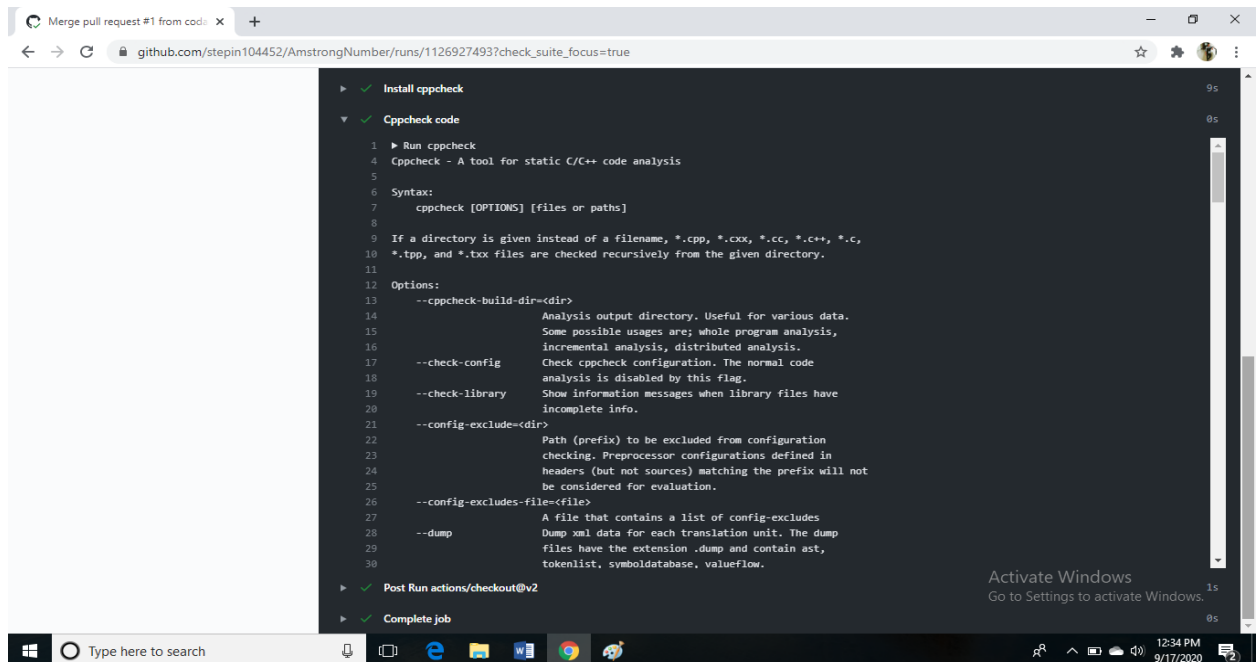
make all 4s

```
1 Run make all
2 make all
3 shell: /bin/bash -e {0}
4 mkdir build
5 gcc main.c src/AmstrongNumber.c -Iinc -o build/amstrong_number_check.out.out
```

Post Run actions/checkout@v2 0s

Activate Windows
Go to Settings to activate Windows.

4) CPPCHECK



Merge pull request #1 from codacy-badger/codacy-ba...
github.com/stepin104452/AmstrongNumber/runs/1126927493?check_suite_focus=true

Install cppcheck 9s

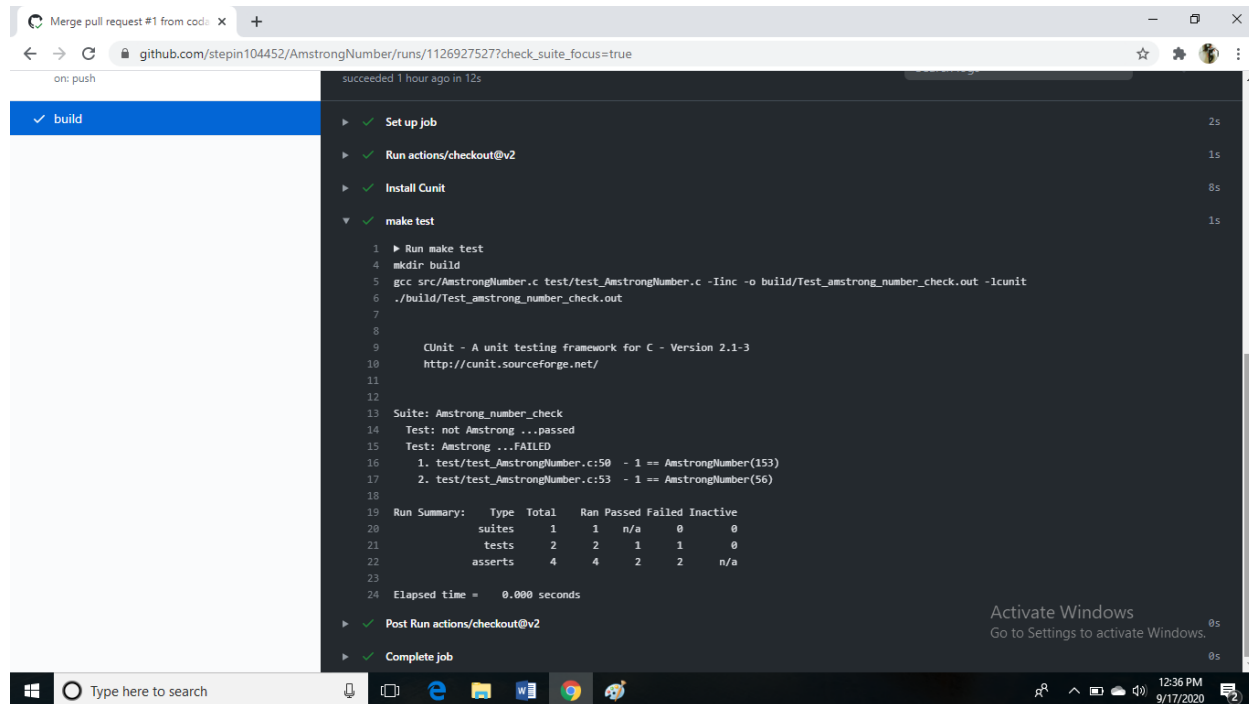
Cppcheck code 0s

```
1 Run cppcheck
4 Cppcheck - A tool for static C/C++ code analysis
5
6 Syntax:
7   cppcheck [OPTIONS] [files or paths]
8
9 If a directory is given instead of a filename, *.cpp, *.c, *.cc, *.c++, *.c,
10 *.hpp, and *.h files are checked recursively from the given directory.
11
12 Options:
13   --cppcheck-build-dir=<dir>
14     Analysis output directory. Useful for various data.
15     Some possible usages are: whole program analysis,
16     incremental analysis, distributed analysis.
17   --check-config
18     Check cppcheck configuration. The normal code
19     analysis is disabled by this flag.
20   --check-library
21     Show information messages when library files have
22     incomplete info.
23   --config-exclude=<dir>
24     Path (prefix) to be excluded from configuration
25     checking. Preprocessor configurations defined in
26     headers (but not sources) matching the prefix will not
27     be considered for evaluation.
28   --config-excludes-file=<file>
29     A file that contains a list of config-excludes
30     files have the extension .dump and contain ast,
31     tokenlist, symbol database, valueflow.
32
33 Post Run actions/checkout@v2 1s
```

Complete job 0s

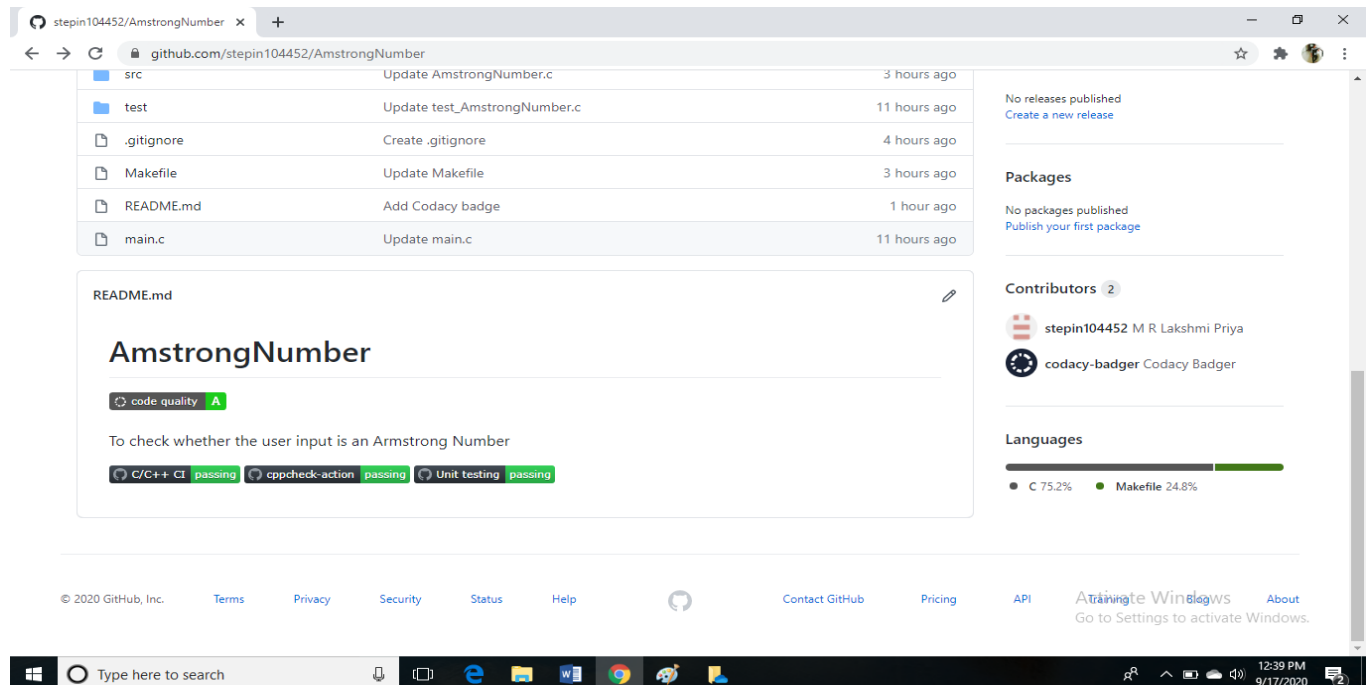
Activate Windows
Go to Settings to activate Windows.

5) UNIT TESTING



```
1  Run make test
4  mkdir build
5  gcc src/ArmstrongNumber.c test/test_ArmstrongNumber.c -Iinc -o build/Test_armstrong_number_check.out -lcunit
6  ./build/Test_armstrong_number_check.out
7
8
9  CUnit - A unit testing framework for C - Version 2.1-3
10 http://cunit.sourceforge.net/
11
12
13 Suite: Armstrong_number_check
14 Test: not Armstrong ...passed
15 Test: Armstrong ...FAILED
16 1. test/test_ArmstrongNumber.c:50 - 1 == ArmstrongNumber(153)
17 2. test/test_ArmstrongNumber.c:53 - 1 == ArmstrongNumber(56)
18
19 Run Summary: Type Total Ran Passed Failed Inactive
20 suites      1      1 n/a      0      0
21 tests       2      2      1      1      0
22 asserts     4      4      2      2 n/a
23
24 Elapsed time = 0.000 seconds
```

6) BADGES



src Update ArmstrongNumber.c 3 hours ago

test Update test_ArmstrongNumber.c 11 hours ago

.gitignore Create .gitignore 4 hours ago

Makefile Update Makefile 3 hours ago

README.md Add Codacy badge 1 hour ago

main.c Update main.c 11 hours ago

README.md

AmstrongNumber

code quality **A**

To check whether the user input is an Armstrong Number

C/C++ CI **passing** cppcheck-action **passing** Unit testing **passing**

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

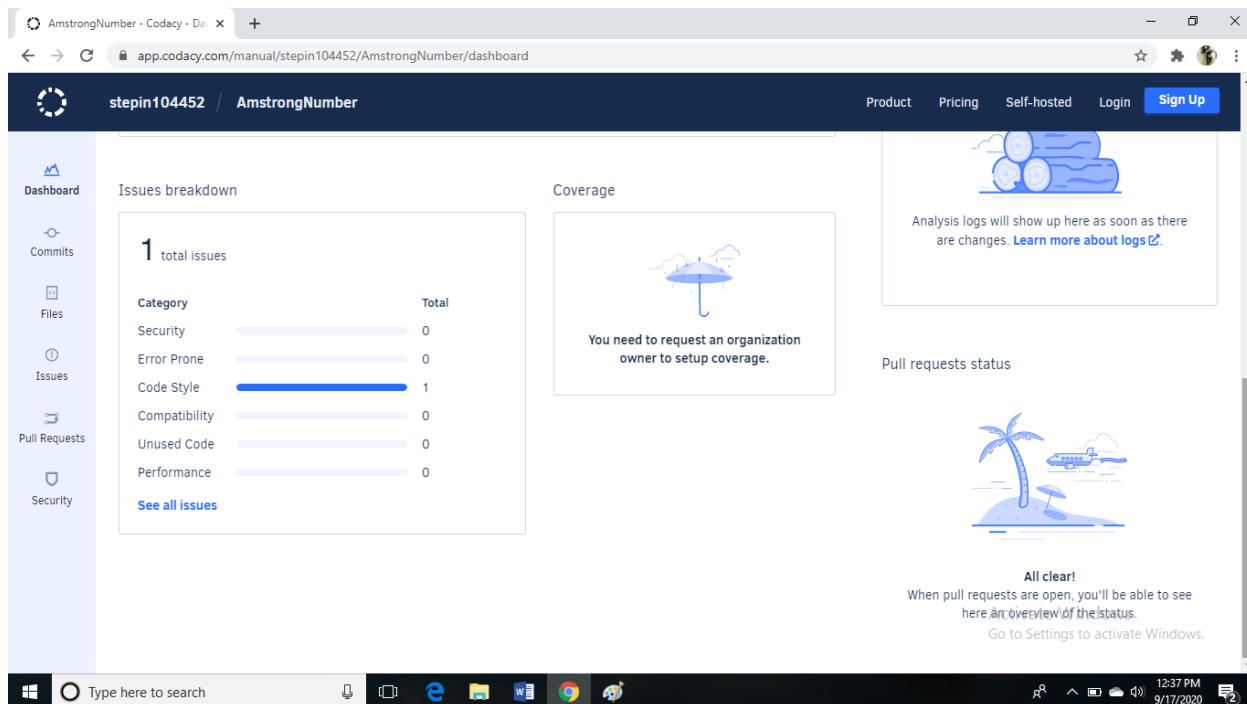
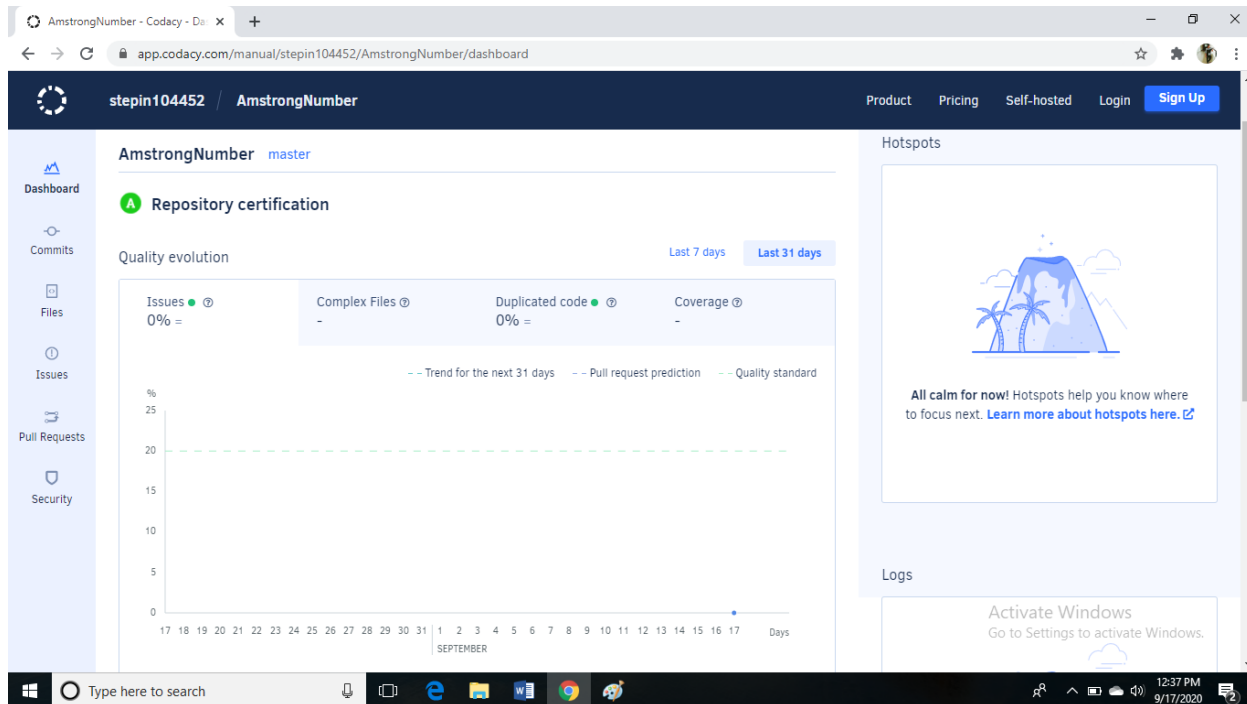
Contributors 2

- stepin104452 M R Lakshmi Priya
- codacy-badger Codacy Badger

Languages

C 75.2% Makefile 24.8%

7) CODACY



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 AmstrongNumber(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 ArmstrongNumber(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      = -linc
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
```