./

Learning Report – Applied SDLC and Software Testing

Course Code: <CODE>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **Approved By** | **Remarks/Revision Details** |
| 1 | 16-09-2020 | Shiva Kumar | Rajini R and Madhu K | Prithvi Sekhar Pagala, Srinivas k | VTC requirements ,design model test plan |
| 2 | 17-09-2020 | Shiva kumar | Rajini R and Madhu K | Prithvi Sekhar Pagala, Srinivas k | Agile aspects |
| 3 | 18-09-2020 | Shiva kumar | Rajini R and Madhu K | Prithvi Sekhar Pagala, Srinivas k | Project-smart calculator-requirements, design, uml diagrams, test plan, user stories |
|  |  |  |  |  |  |
|  |  |  |  |  |  |



**Document History**

# 

Contents

[Activity 1– System/Software Development 4](#_Toc52053888)

[Research: 4](#_Toc52053889)

[Ageing: 4](#_Toc52053890)

[Requirements: 4](#_Toc52053891)

[Specific requirements: 5](#_Toc52053892)

[Software Requirements 9](#_Toc52053893)

[Functional Requirement: 9](#_Toc52053894)

[Non-Functional Requirement: 10](#_Toc52053895)

[DESIGN: 10](#_Toc52053896)

[High level design: 10](#_Toc52053897)

[Low level Design: 12](#_Toc52053898)

[Test Plan**:** 13](#_Toc52053899)

[Requirements Based test plan: 13](#_Toc52053900)

[Scenario Based test plan: 14](#_Toc52053901)

[Boundary conditions: 14](#_Toc52053902)

[TEST PLAN TABLE: 15](#_Toc52053903)

[REFERENCES 16](#_Toc52053904)

[AGILE ACTIVITY for Activity -1: 16](#_Toc52053905)

[THEME: 16](#_Toc52053906)

[EPIC: 16](#_Toc52053907)

[CASE STUDIES: 16](#_Toc52053908)

[Activity 2 –CI Workflow for C Programming 17](#_Toc52053909)

[GIT: 17](#_Toc52053910)

[COMMITS: 17](#_Toc52053911)

[MAKEFILE: 18](#_Toc52053912)

[BUILD: 18](#_Toc52053913)

[CPPCHECK: 19](#_Toc52053914)

[UNIT TESTING: 19](#_Toc52053915)

[CODE QUALITY: 20](#_Toc52053916)

[GITHUB Link: 21](#_Toc52053917)

[ACTIVITY 3-MINI PROJECT 21](#_Toc52053918)

[Define product: 21](#_Toc52053919)

[Ageing: 21](#_Toc52053920)

[Cost and Gradation 21](#_Toc52053921)

[Requirements: 21](#_Toc52053922)

[High-level and Low-level requirements: 22](#_Toc52053923)

[Design: 22](#_Toc52053924)

[High level design: 23](#_Toc52053925)

[Low level design: 24](#_Toc52053926)

[Test Plan: 25](#_Toc52053927)

[SWOT Analysis: 26](#_Toc52053928)

[Activity 3 – Agile Aspects 26](#_Toc52053929)

[Theme: 27](#_Toc52053930)

[Epic: 27](#_Toc52053931)

[User Stories: 27](#_Toc52053932)

Table for figures and Tables

[Figure 1 requirement table for vehicle tracking system 6](#_Toc52098724)

[Figure 2 user interface for vehicle tracking system 7](file:///C:\Users\shiva%20kumar\Downloads\LearningReport_GENESIS%20(1).docx#_Toc52098725)

[Figure 3 ATmega328 features 7](#_Toc52098726)

[Figure 4 GPS Antenna 10](#_Toc52098727)

[Figure 5 GSM Antenna 10](#_Toc52098728)

[Figure 6 Arduino Software interface 11](#_Toc52098729)

[Figure 7 overview of vehicle tracking system design model 12](file:///C:\Users\shiva%20kumar\Downloads\LearningReport_GENESIS%20(1).docx#_Toc52098730)

[Figure 8 sequence diagram for designed model 13](#_Toc52098731)

[Figure 9 component diagram for the system 13](#_Toc52098732)

[Figure 10 top level design system 14](#_Toc52098733)

[Figure 11 activity diagram for top level working system 14](file:///C:\Users\shiva%20kumar\Downloads\LearningReport_GENESIS%20(1).docx#_Toc52098734)

[Figure 12 flow chart for working system 15](#_Toc52098735)

[Figure 13 table of test plan for vehicle tracking system 17](#_Toc52098736)

[Figure 14 git repo for CI workflow for C programming 19](#_Toc52098737)

[Figure 15 commits in C programming 20](#_Toc52098738)

[Figure 16 make file for C programming 21](#_Toc52098739)

[Figure 17 screenshot of Build 21](#_Toc52098740)

[Figure 18 screenshot of CPPCHECK 22](#_Toc52098741)

[Figure 19 screenshot of unit testing 22](#_Toc52098742)

[Figure 20 screenshot of codacy badge 23](#_Toc52098743)

[Figure 21 table of requirements for smart calculator 25](#_Toc52098744)

[Figure 22 high level and low level requirement for smart calculator 25](#_Toc52098745)

[Figure 23 activity diagram for smart calculator 26](#_Toc52098746)

[Figure 24 class diagram for smart calculator 27](#_Toc52098747)

[Figure 25 sequence diagram for smart calculator 28](#_Toc52098748)

[Figure 26 table of test plan for smart calculator 29](#_Toc52098749)

[Figure 27 table of SWOT analysis 30](#_Toc52098750)

# Activity 1– System/Software Development

**VEHICLE TRACKING SYSTEM**

## Research:

In vehicle tracking system Arduino Uno is used because earlier vehicle tracking system was based on very complicated and high cost microcontroller like 8051 with programming kit, ARM, etc.

The main advantage of Arduino is low cost and easy to program it also compatible with all the three operating system(windows,mac,linux) that’s why Arduino is better and further information is given below, Arduino is an open source, computer hardware and software company, project, and user community that designs and manufactures microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world.

## Ageing:

Earlier GPS navigation was only used by the military purposes but later date every individual can use GPS that is embedded in their smart phones. The American Public Transportation Association estimated that, at the beginning of 2009, around half of all transit buses in the United States were already using a GPS-based vehicle tracking system to trigger automated stop announcements.

Now a days Vehicle tracking systems are also popular in consumer vehicles as a theft prevention, monitoring and retrieval device.

## Requirements:

|  |  |
| --- | --- |
| **ID NO** | **DESCRIPTION** |
| **H\_01** | GSM module |
| **H\_02** | GPS module |
| **H\_03** | Microcontroller |
| **H\_04** | Batteries |
| **H\_05** | GSM antenna |
| **H\_06** | GPS antenna |
| **H\_01\_02\_L\_01** | Sim808 module is the combination of both GSM/GPRS and GPS navigation system with Control via AT commands (3GPP TS 27.007,27.005 and SIMCOM enhanced AT Commands) |
| **H\_03\_L\_02** | Arduino Uno is a microcontroller belong to atmel family. |
| **H\_04\_L\_03** | 12 Volts battery with 2A current |
| **H\_05\_L\_04** | Operating at both GSM Quad Band Frequencies with +2dBi gain.  This antenna operates in Quad Band 890/960, 1710/1880 MHz Frequencies |
| **H\_06\_L\_05** | Operating frequency range is 1575.42±1.023 MHz and voltage range is 2.5V- 5.5V and corresponding current range is 6.6 mA - 16.6 mA |

Figure 1 requirement table for vehicle tracking system

### Specific requirements:

**External interface requirements:**

**User Interfaces:**

Message for location:

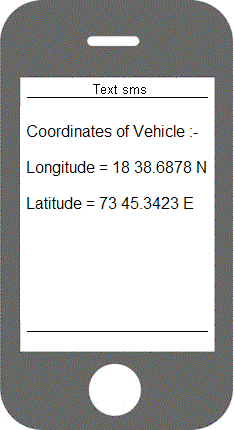
latitude: 2400.0090, N longitude: 12100.0000, E time: 12:00’’

Figure user interface for vehicle tracking system

Hardware interfaces:

MICROCONTROLLER ATMEGA328:

Arduino Uno is based on ATMega328. ATMega328 is the ATMEL Microcontroller. It is a single chip microcontroller and belongs to the mega AVR family.

The Atmega328 provides the following features:

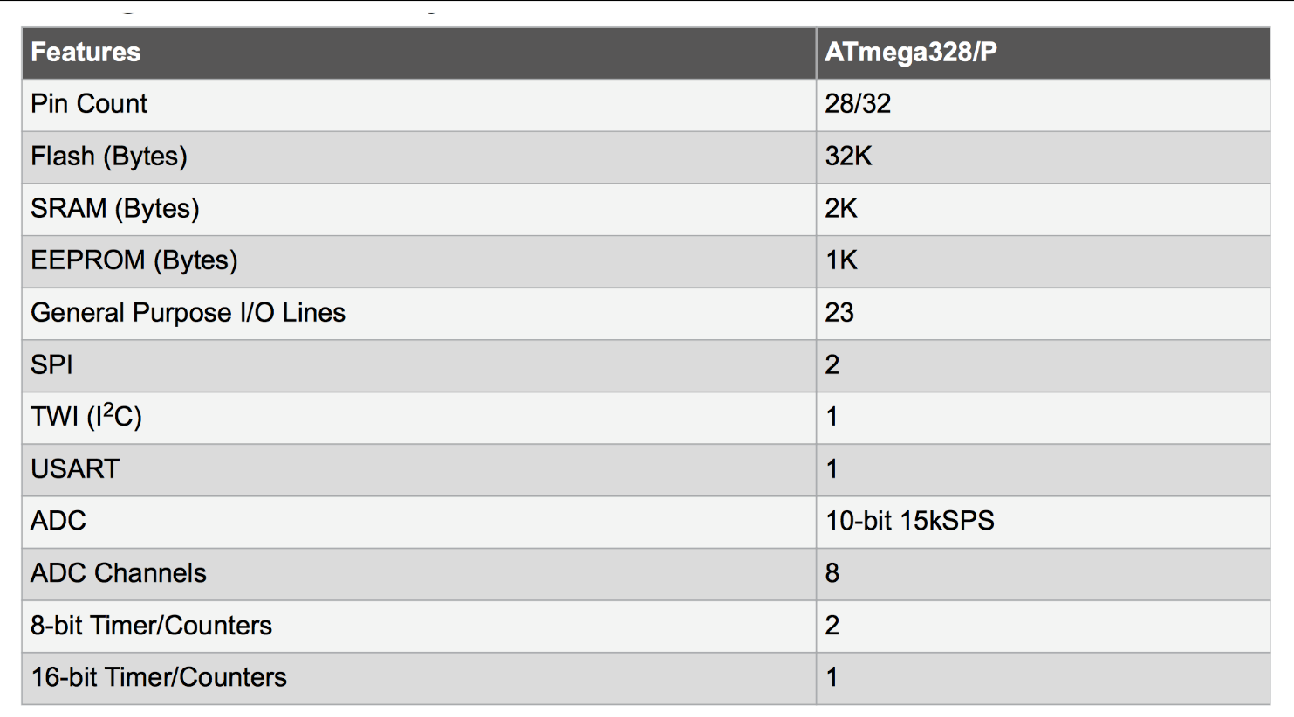


Figure 3 ATmega328 features

The Atmega328 has 28 pins.

14 pins are digital of which 6 can give PWM output and 6 can give analog input/output. **PWM** is a technique for getting analog results. 2 pins are crystal oscillator, which are used as a clock pulse. These I/O pins account for 20 of the pins.

The Atmega328 chip has an analog-to-digital converter (ADC) inside of it. The

microcontroller needs power so 2 pins VCC and GND are dedicated, so that it can operate. The power required by the Atmega328 is 1.8-5.5V of power operate.

Because there is an **ADC**, the chip can interpret analog input, which is why the chip has 6 pins for analog input.

Analog voltages higher than the reference voltage will be assigned to a digital value of 1, while analog voltages below the reference voltage will be assigned the digital value of 0. The last pin is the RESET pin. This allows a program to be rerun and start over.

**SIM808 MODULE:**

SIM808 module is a GSM and GPS two-in-one function module. It is based on the latest GSM/GPS module SIM808 from SIMCOM, supports GSM/GPRS Quad-Band network and combines GPS technology for satellite navigation.

The latest firmware support bluetooth function. You can use all AT commands for BT Functions right now. It features ultra-low power consumption in sleep mode and integrated with charging circuit for Li-Ion batteries, that make it get a super long standby time and convenient for projects that use rechargeable Li-Ion battery. It has high GPS receive sensitivity with 22 tracking and 66 acquisition receiver channels. Besides, it also supports A-GPS that available for indoor localization. The module is controlled by AT command via UART and supports 3.3V and 5V logical level.

**FEATURES:**

* Quad-band 850/900/1800/1900MHz
* GPRS multi-slot class12 connectivity: max. 85.6kbps(down-load/up-load)
* GPRS mobile station class B
* Controlled by AT Command (3GPP TS 27.007, 27.005 and SIMCOM enhanced AT Commands)
* Supports charging control for Li-Ion battery
* Supports Real Time Clock
* Supply voltage range 3.4V ~ 4.4V
* Integrated GPS/CNSS and supports A-GPS
* Supports 3.0V to 5.0V logic level
* Low power consumption, 1mA in sleep mode
* Supports GPS NMEA protocol
* Standard SIM Card

**GPS Antenna:**

This GPS antenna draws about 10mA and will give you an additional 28 dB of gain. It got

a 5 meter long cable so it will easily reach wherever it is needed to. The antenna is magnetic

so it will stick to the top of a car or truck or any other steel structure. Its operating frequency

range is 1575.42±1.023 MHz and voltage range is 2.5V- 5.5V and corresponding current

range is 6.6 mA - 16.6 mA. GPS signals are extremely weak and present unique demands on the antenna so the choice of antenna plays an important role in GPS performance. A GPS unit needs to have a clear, unobstructed sky view, to best receive the microwave signals that allow it to communicate with satellites. GPS Down/Up converter used for very long cable runs. This GPS antenna that receives the GPS signal, converts it to a lower frequency which is then sent down the cable. Next to the GPS receiver is an up converter that converts the signal back to the original frequency and delivers it to the GPS receiver.



Figure 4 GPS Antenna

**GSM Antenna:**

GSM communications are dependent on antennas. The antenna is what allows

communications signals to be sent and received. The antenna that we have used in our project provides operation at both GSM Quad Band Frequencies with +2dBi gain.

This antenna operates in Quad Band 890/960, 1710/1880 MHz Frequencies and

it’s an omni-directional.



Figure 5 GSM Antenna

Software Requirements:

The Arduino IDE is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring project. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. There is typically no need to edit make files or run programs on a command-line interface. Although building on command-line is possible if required with some third-party tools such a Ino.

The Arduino IDE comes with a C/C++ library called "Wiring" (from the project of the same name), which makes many common input/output operations much easier. Arduino programs are written in C/C++.

****

Figure 6 Arduino Software interface

### Functional Requirement:

Whenever a vehicle is parked, it is kept at vehicle tracking security mode. Once the vehicle is stolen, the position of the vehicle changes, the owner can send an SMS to the vehicle to know the location or position of the vehicle. The SMS sent would pass through the GSM service provider and then reach the vehicle, which is travelling, because the vehicle has a GSM device with a SIM card. This GSM modem will receive the SMS and send to the microcontroller in the vehicle. The microcontroller will receive this SMS and compare the password and the command. If the information matches the already programmed one, then it will perform the request required by the owner. It will then send the required location; latitude, longitude and time to the registered number of the owner and the results will be display on the screen of the owner’s mobile phone.

### **Non-Functional Requirement:**

* + - * Access to the location of the vehicle should be restricted to people that are authenticated to view information.
      * Passwords and ID’s should be regulated to be at least a certain length and must contain non-alphanumeric characters in both the password and ID to make the system more secure.
      * The device should be installed in such a way that it can’t be affected by any atmospheric factors such as rain.

## DESIGN:

### High level design:

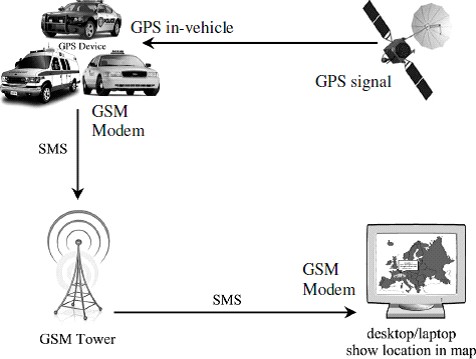


Figure overview of vehicle tracking system design model

SEQUENCE DIAGRAM:

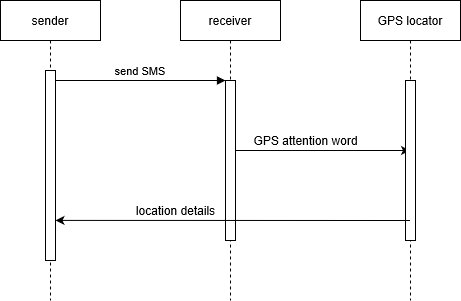


Figure 8 sequence diagram for designed model

COMPONENT DIAGRAM:

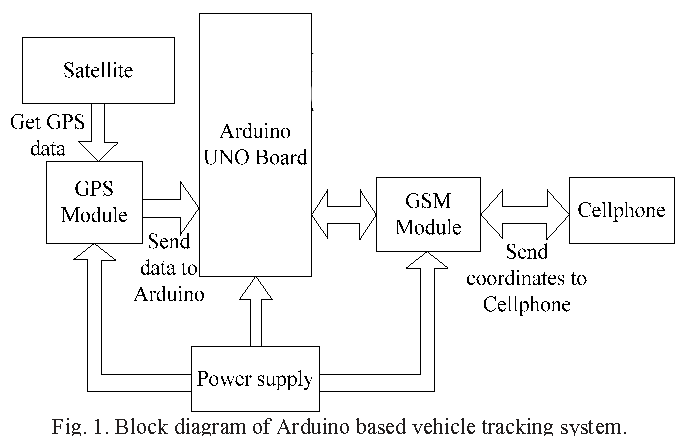


Figure 9 component diagram for the system

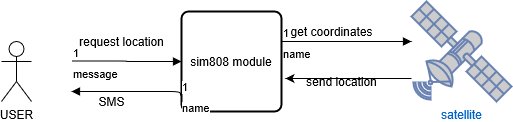


Figure 10 top level design system

### Low level Design:

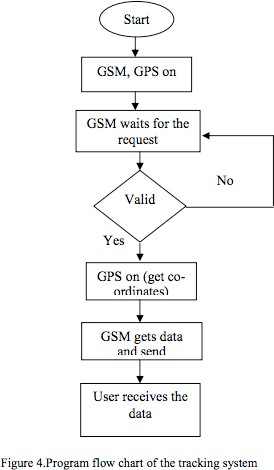
**ACTIVITY DIAGRAM:**

Figure activity diagram for top level working system

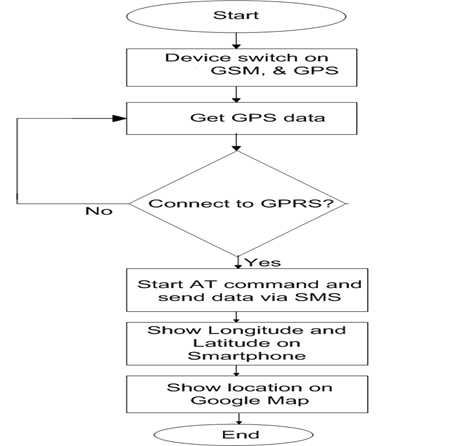


Figure 12 flow chart for working system

## Test Plan**:**

### Requirements Based test plan:

* Proper functioning of a GPS receiver requires undisturbed reception of signals from at least four GPS satellites.
* In any measuring system, there is a limit to the accuracy with which measurements can be made. GPS module gives an error of 5-10 meters.
* In order to determine the route of the vehicle, we need a storage device. We need to provide an external storage unit to store the coordinates of the entire journey.
* GPS modem is very power consuming.
* The location of the vehicle can be seen only on the registered mobile number. Hence the user must update the changes in the mobile number.

### Scenario Based test plan:

* The main users of the product are the transportation companies. Companies and start-ups that provide cabs can install the device in their vehicles to ensure the safety of passengers.
* Other users could be traffic police, who can view the traffic on the roads and can effectively manage the existing traffic by diverting them to another route.
* Schools could also install the device in school vans can keep a constant check on the route of the vans for the sake of students’ safety.
* It could be used in Bus stands by the managers to effectively predict and display the timing of the arriving buses for the waiting passengers.

### Boundary conditions:

* **Signal Propagation Error:** We assume that atmosphere has no effect on the signal sent by the satellites to the GPS module. Such an error is called spherical error.
* **Receiver’s Error:** We assume that there is no receiver clock error and measurement noise. Under worst case scenarios, the receiver clock error could be very large.
* We assume that the mobile number that user will provide is not going to change in the future.
* It is assumed that the user has the basic knowledge of using a mobile in order to use the product effectively.
* The functioning of the product depends on the quality of the GPS, strength of GPS signal, GSM module, the signal strength of the mobile network and the coverage area of the company’s SIM.

## TEST PLAN TABLE:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Description | Precondition | Expected input | Expected output | Actual output |
| H\_01\_02 | SIM80 module combination of GPS and GSM module | Initially GPS module should able to work | Sim808 module getting 5v supply | Sim808 is set to text mode for GSM communication | Sim808 is set to text mode for GSM communication |
| 2 | Arduino uno | Arduino uno require 5v input as VCC and ground | Making Arduino to work as sender and receiver | Arduino is consider as the brain for this system | Arduino need to be be working properly |
| 3 | Battery | Battery supply of 12 v and 2.1A | Bettery supply is perfectly working | Bettery supply is perfectly working | Bettery supply is perfectly working |

Figure 13 table of test plan for vehicle tracking system

## REFERENCES

* Dr. Ravi Vijay, Dilip Sharma, Sakshi Sharma, Deepak Mali, Rajesh Solanki “VEHICLE TRACKING SYSTEM USING GPS AND GSM”, Dept. of Electronics and Communication Eng, Ujjain Engineering College, Ujjain (M.P.), 2014.
* Krishna Kant, “Microprocessor and microcontroller”, Eastern Company Edition, New Delhi 2007. Prawat Chairprapa, Supaporn Kiattisin and Adisorn Leelasantitham “A Real-Time GPS Vehicle Tracking System Displayed on a Google Map Based Website”, Computer and Multimedia Engineering, School of Engineering University of the Thai Chamber of Commerce, 2004.
* Kunal Maurya, Mandeep Singh, Neelu Jain “Real Time Vehicle Tracking System using GSM and GPS Technology – An Anti-theft Tracking System”, Department of Electronics, PEC Punjab University of Technology, Chandigarh, 2013

## AGILE ACTIVITY for Activity -1:

### THEME:

System embedded with gsm and gps module is able to track the exact location of the vehicle with accurate coordinates is made available to the user with registered mobile number and google maps link send to the user when requested.

### EPIC:

* Controlling unit
* Batteries
* GSM module
* GPS module
* Mobile phone with internet connectivity

### CASE STUDIES:

* As a **Wildlife researchers and animal conservation agencies** can track the movement of animals and their migratory patterns. The systems can be attached to their collars ,harness or put directly on their bodies.
* As a **Parent** we can track the movement and activities of the child.
* **As a Ambulance Tracking and Emergency Medical Services Fleet**  
  Through GPS vehicle tracking system, ambulances and EMS Fleet can be monitored and sent quickly to the critical patients, thus providing timely medical facility and saving lives.
* GPS Tracking System helps you to keep a track of your car, luggage, valuable possessions, pets and even loved ones. They can be located easily keeping you free of stress and worries. GPS Tracker can also help you in locating where your child is going late at night.

# Activity 2 –CI Workflow for C Programming

## GIT:

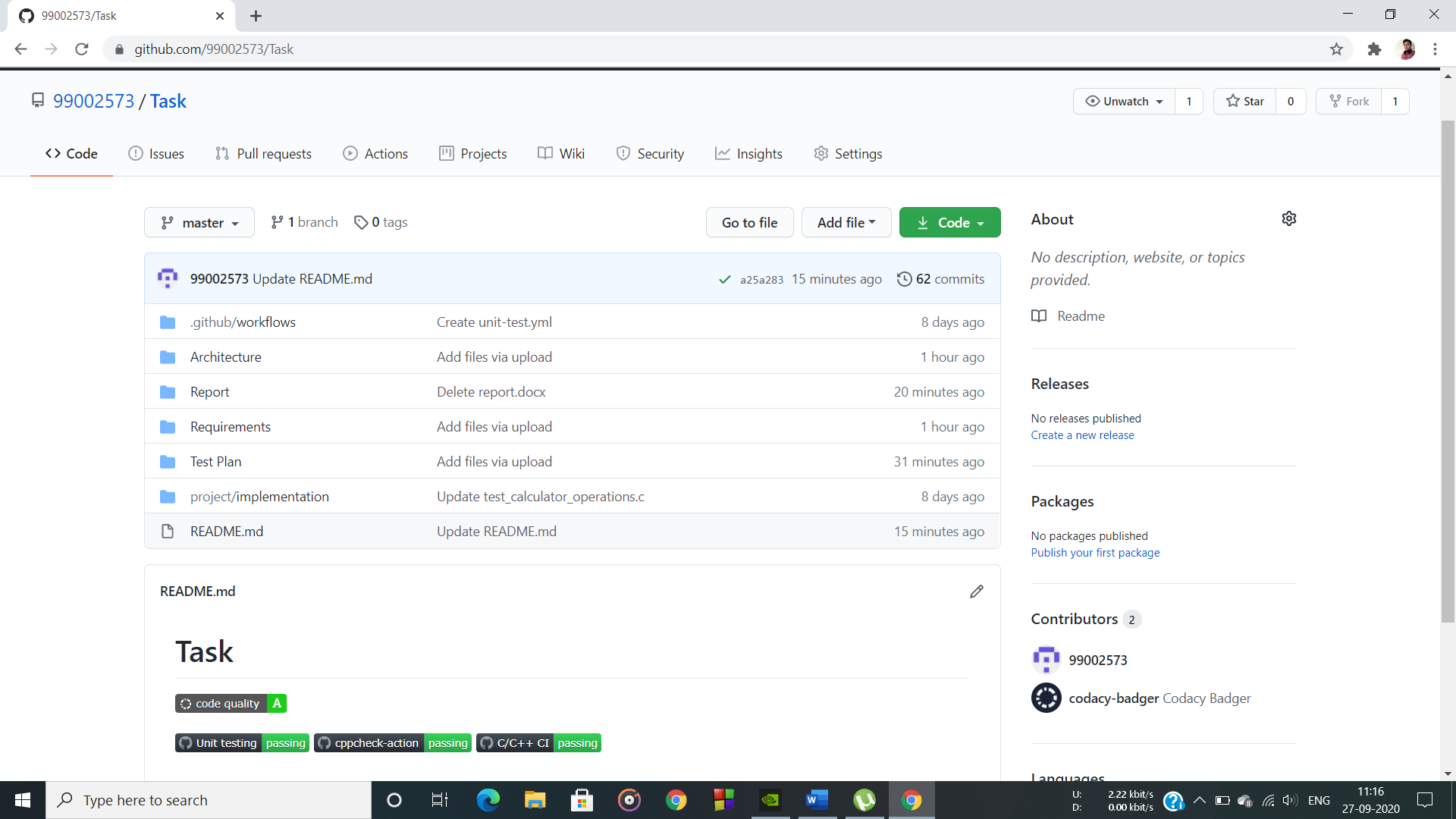


Figure 14 git repo for CI workflow for C programming

## COMMITS:

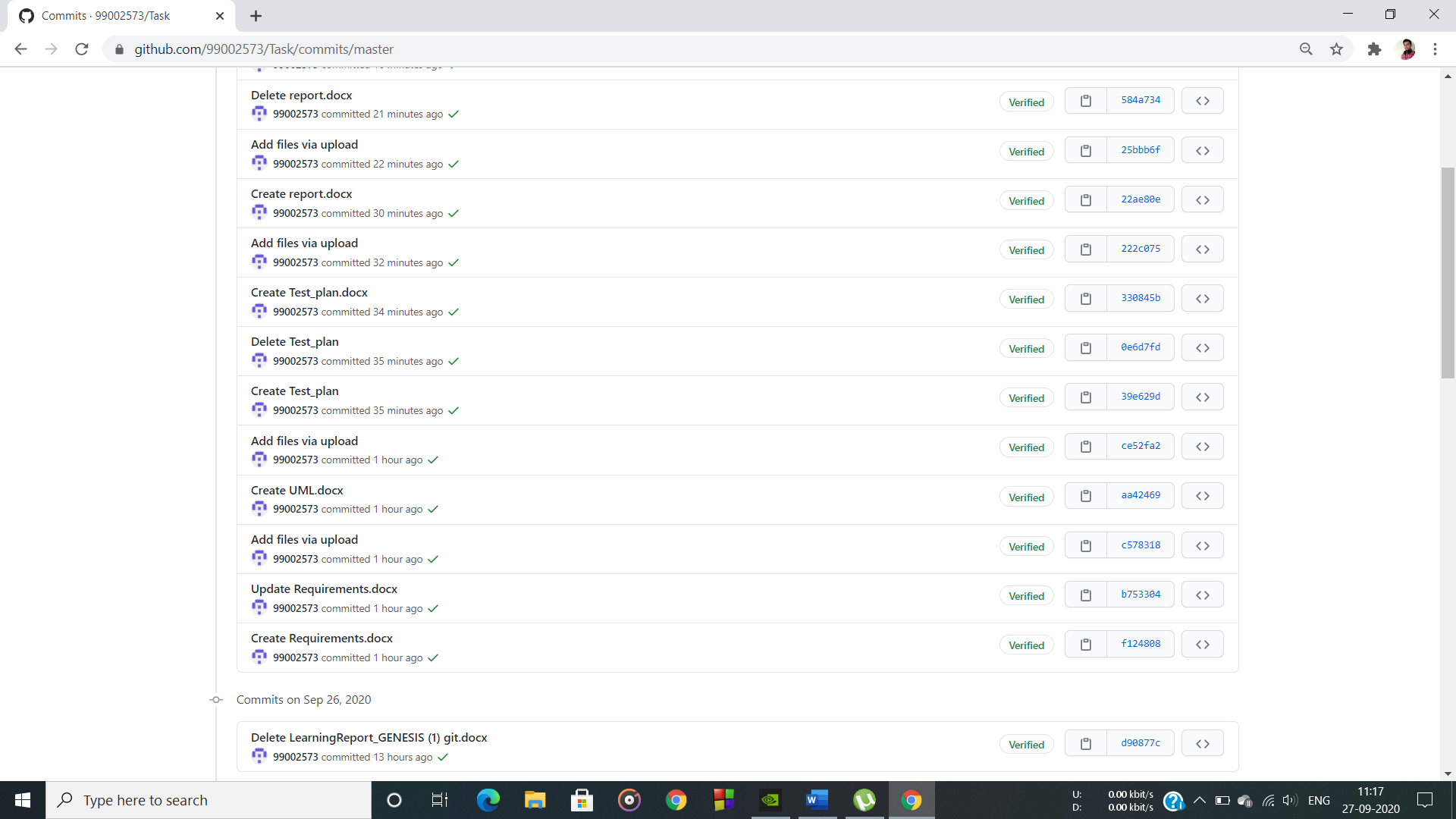


Figure 15 commits in C programming

## MAKEFILE:

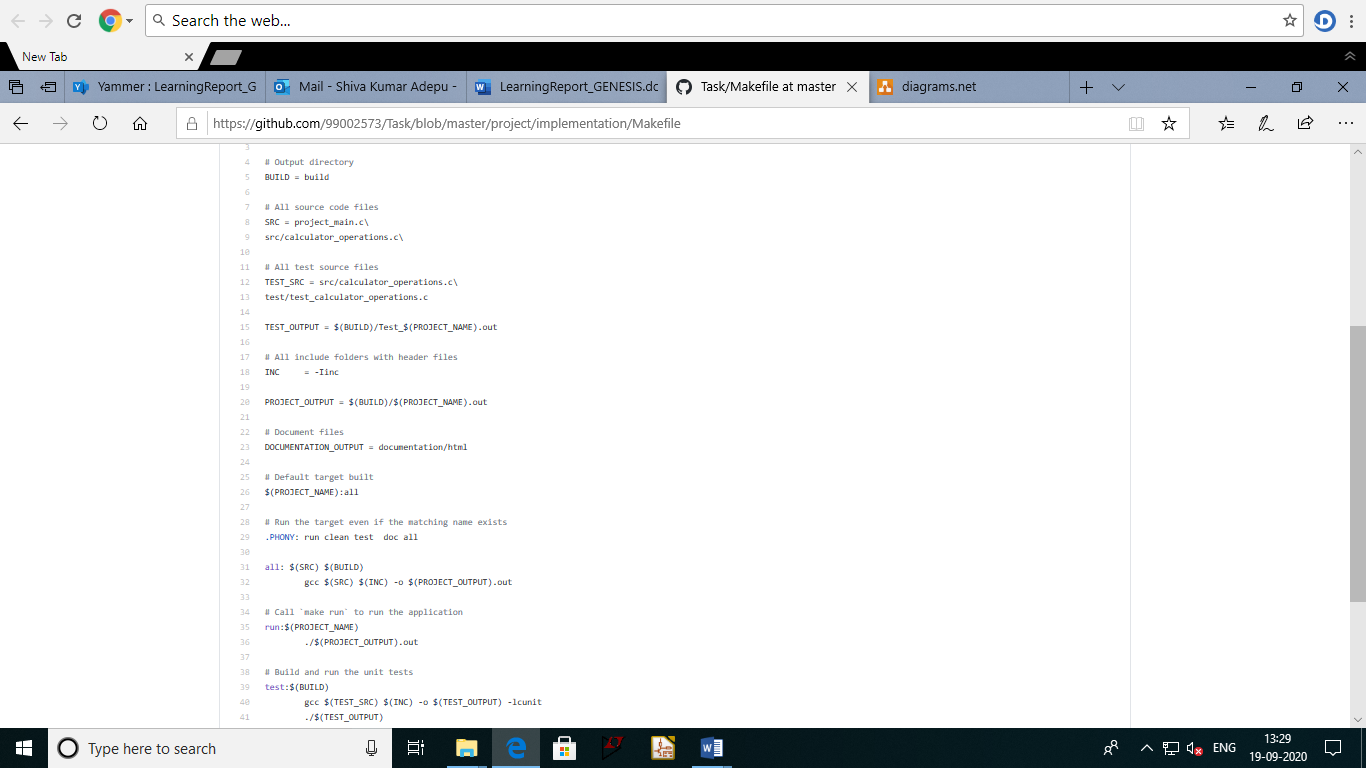


Figure 16 make file for C programming

## BUILD:

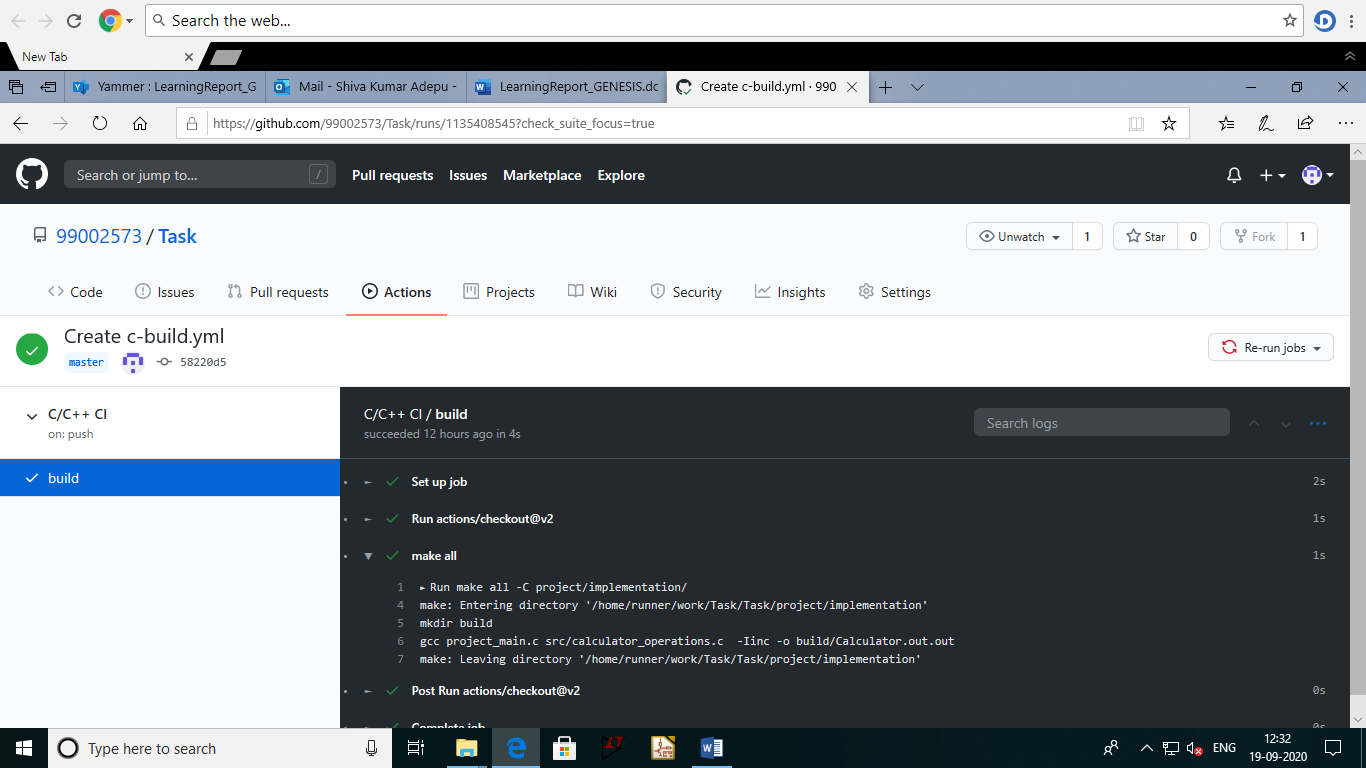


Figure 17 screenshot of Build

## CPPCHECK:

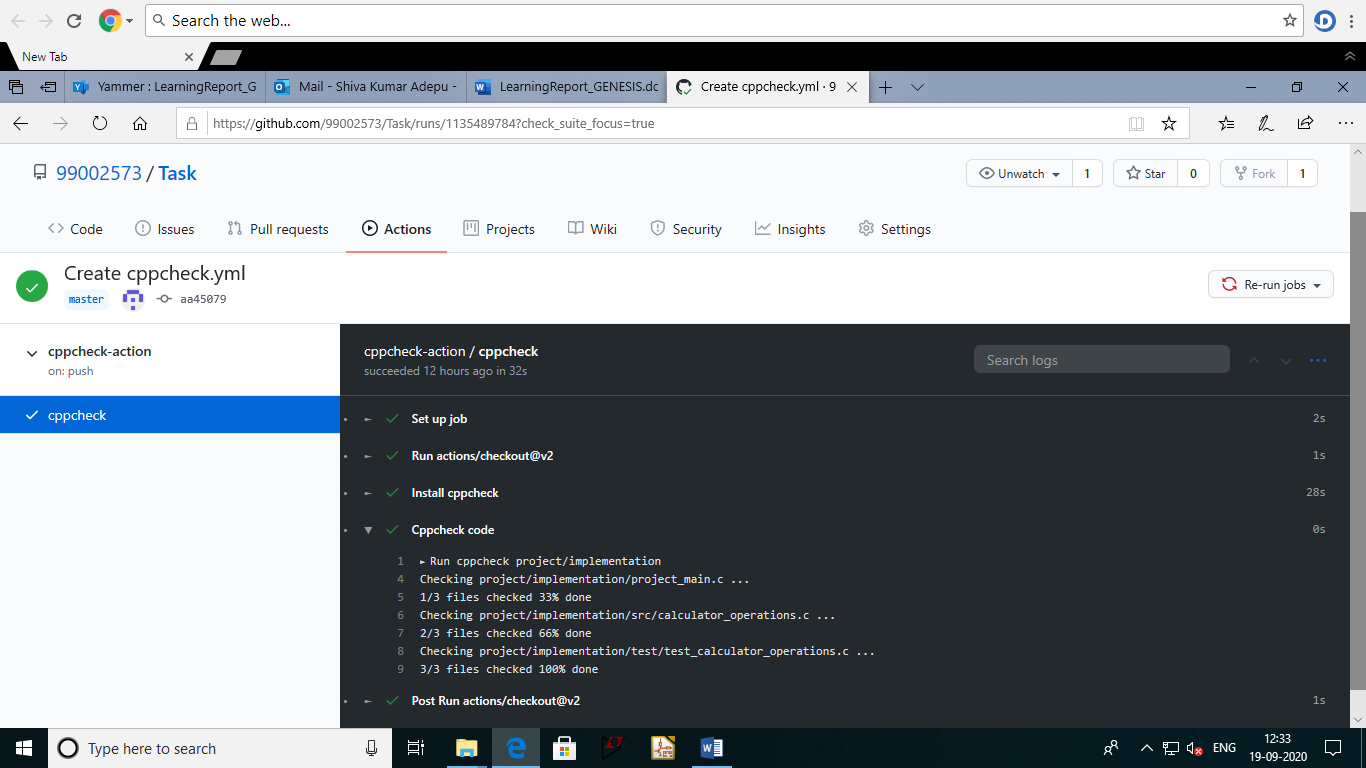


Figure 18 screenshot of CPPCHECK

## UNIT TESTING:

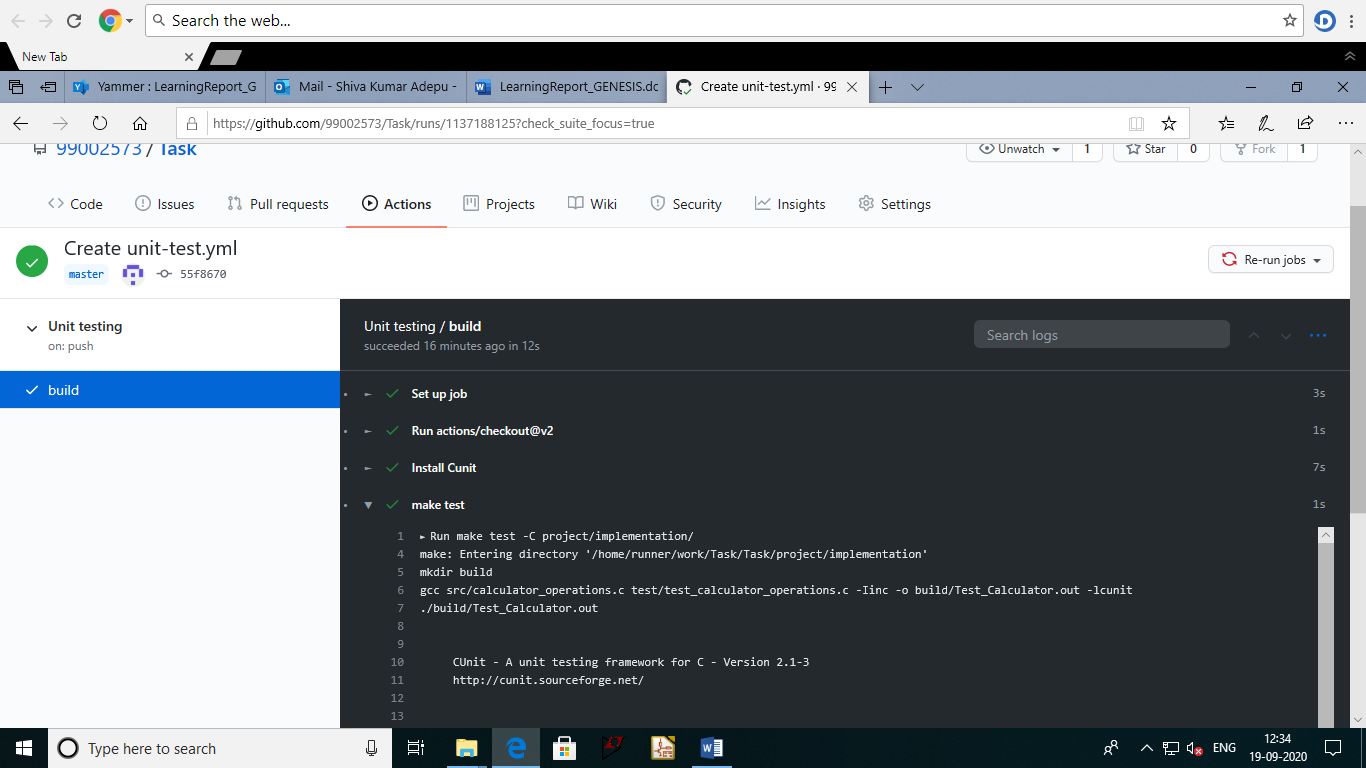


Figure 19 screenshot of unit testing

## CODE QUALITY:

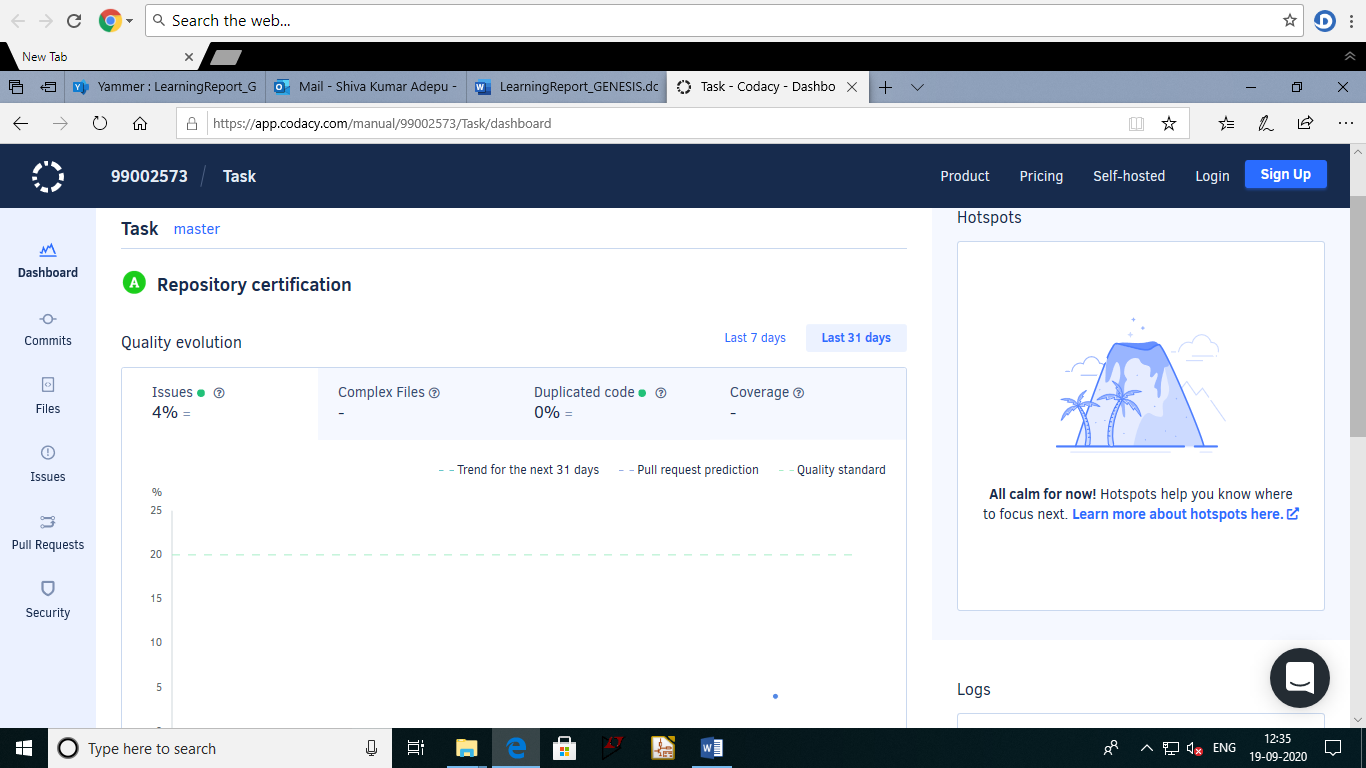


Figure 20 screenshot of codacy badge

## GITHUB Link:

<https://github.com/99002573/Task>

referred: <https://github.com/stepin654321/MiniProject_Template/tree/master/MiniProject_C>

# ACTIVITY 3-MINI PROJECT

PRODUCT – SMART CALCULATOR

## Define product:

This document specifies the functional requirements for a multi-function calculator program. The program is designed to act like a “handheld calculator” with the usual standard functions (add, subtract, multiply, divide). Additionally, the calculator will have the additional capability of performing functions like squares, cubes, power, maximum, factorial, check prime, average, swap. The program is designed to be as easy to use as a regular calculator.

## Ageing:

**Past:**

Mechanical calculators are used to perform basic arithmetic operations and they are comparable in size to small desktop computers.

**Present:**

Digital and scientific calculator are portable and used to perform calculations ranging from basic arithmetic to complex mathematics. Scientific calculators include trigonometric and statistical calculations.

**Future:**

Voice command input calculators can be developed.

Cost and Gradation**:**

Product cost will increase for every new technology but as the new technologies evolve, the product cost reduces.

Cost of standard calculators: Rs.300-Rs.500

Cost of digital calculators: Rs.500-Rs.800

Cost of scientific calculators: Rs.800-Rs.1000

## Requirements:

|  |  |
| --- | --- |
| ID | DESCRIPTION |
| 1 | The calculator has the following keys 0-9, +, -, \*, /, =, DEL, AC. and operations keys like power, squares, cubes, max, factorial, average, prime, swap. |
| 2 | In any situation, calculator must produce a correct result defined by the well-known arithmetic rules. |
| 3 | If the calculations are impossible the calculator must display information helping the user to resolve the erroneous situation |
| 4 | On encountering a division by 0 the display should read "Math Error" and typing the key “AC” should reset the calculator |
| 5 | On calculating the “factorial” value of a negative operand, the display should read "Wrong operand". |
| 6 | For operation key ‘max’ the fucntions identifies the maximum of two operands and display the highest value operand it is possible for both positive and negative operands |
| 7 | For operation key “swap” the two operands swaps their values. |
| 8 | On identifying a non-prime value, the display should read "Not a prime". |
| 9 | On erroneous operand or operation keys the display should read “Reset (AC) to continue” or “Clear (DEL) to continue” as appropriate. Of course, any situation can be cleared using the main reset key “AC”. |
| 10 | The program shall be written in standard C, as compiled by the GNU GCC compiler (). The program shall use only standard C library functions. The program shall be usable on any system which supports the compiler, and shall not require any particular hardware or software. |

Figure 21 table of requirements for smart calculator

### High-level and Low-level requirements:

|  |  |
| --- | --- |
| H\_01\_L\_01 | The program must have the required keys in their respective types-hence impossible results must be ask by showing some hint or symbols. |
| H\_02\_L\_01 | The correct results must be produced within less amount of time -any device with proper compile shall use this program. |
| H\_03\_L\_01 | The system must have the display to show the respective results |
| H\_04\_L\_01 | Indefinite form |
| H\_05\_L\_02 | Imaginary number |

Figure 22 high level and low level requirement for smart calculator

## Design:

This work was centred on design and implementation of simple scientific calculator. This work mainly focused principally on numbers and arithmetic operation. Interestingly, the end result of simple calculator

System was its ability to process number and operators and provides a useful result.

### High level design:

Activity Diagram:

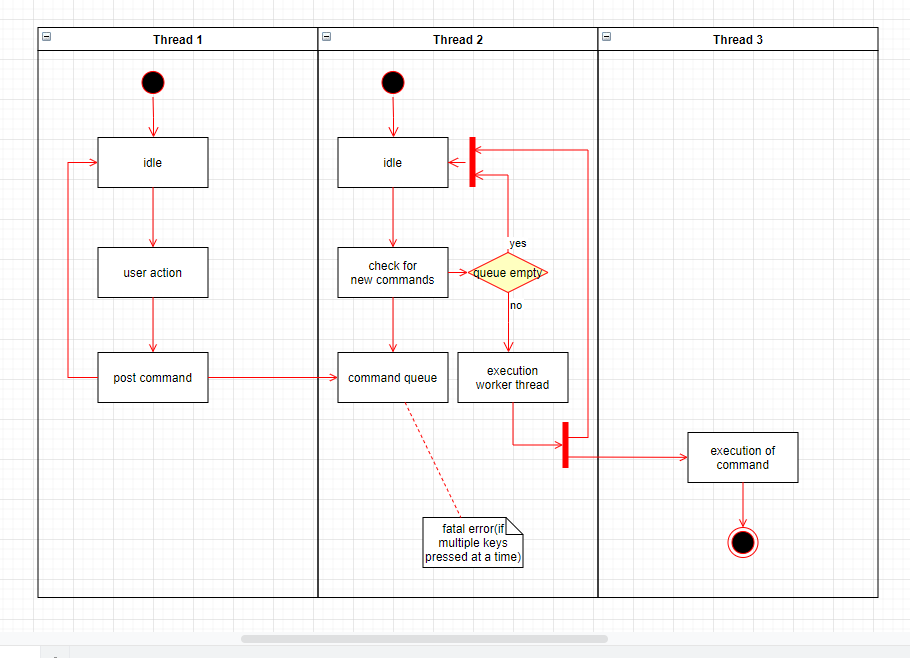


Figure 23 activity diagram for smart calculator

CLASS DIAGRAM:

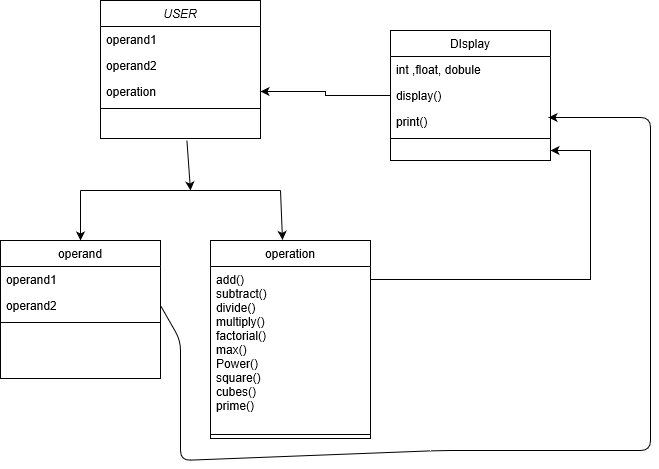


Figure 24 class diagram for smart calculator

### Low level design:

Sequence diagram:

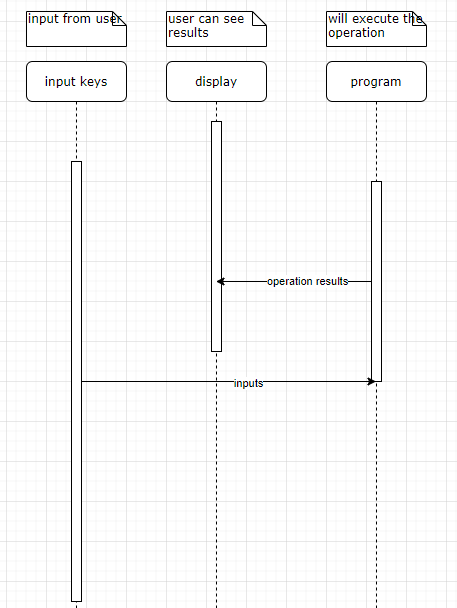


Figure 25 sequence diagram for smart calculator

## Test Plan:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Precondition** | **Expected I/p** | **Expected o/p** | **Actual o/p** |
| High\_1 | Input from user | Accept the input from key | Correct input | Will show the inputs in display | Same as expected o/p |
| High\_2 | Maths operations | Correct Calculation | Correct input symbol as pressed by the user. | Will do correct calculation | Same as expected o/p |
| High\_3 | display | Display output | proper display of output which can be interpreted by human. | will display results clear visibility |  |
| Low\_1 | Format of result | Format of the output can be more than one type | No expected input | Will show error or fatal error | Operation has performed |
| Low\_2.1 | Indefinite number | If the numerator is zero | decimal with number 0 | Error or NA |  |
| Low\_2.2 | Imaginary number | if give the input for root of a negative number. | Root of a negative number | Error or NA |  |
| EX-1 | Addition of 2 numbers | User input | 7 & 3 | 10 | 10 |
| EX-2 | Subtraction of 2 numbers | User input | 5 & 2 | 3 | 3 |
| EX-3 | Division of 2 number s | User input | 3 & 2 | 1 | 1 |
| EX-4 | Multiplication of 2 numbers | User input | 5\*7 | 35 | 35 |
| EX-5 | Square of a number | User input | 8 | 64 | 64 |
| EX-6 | Cubes of a number | User input | 10 | 1000 | 1000 |
| EX-7 | Power of 2 numbers | 1st input is number and 2nd input is exponential factor | 4 & 6 | 4096 | 4096 |
| EX-8 | Factorial of a number | User input | 3 | 6 | 6 |
| Ex-9 | Average of 2 numbers | User input | 5 & 7 | 6 | 6 |
| EX-10 | Swap of 2 numbers | User input | (5,4) | (4,5) | (4,5) |

Figure 26 table of test plan for smart calculator

SWOT Analysis:

|  |  |  |  |
| --- | --- | --- | --- |
| **Strength** | **Weakness** | **Opportunities** | **Threats** |
| Fast calculation | Damaged under 30mm water | Healthy programming practice | Fault result |
| Water proof | not robust and axile | chance to show your skills in c and c++ | Hardware acceleration |
| large calculation | Basic knowledge needed | Business standard ideas | Imaginary and indefinite number or infinite number display problem |
| All arithmetic operations | Need to know prerequisites | Arithmetic operations and logics |  |

Figure 27 table of SWOT analysis

Activity 3 – Agile Aspects

### Theme:

**T1. Accounting.**

**T2. Student.**

### Epic:

**E1.**

* Simple arithmetic operations and large calculations.
* Addition
* Division
* Subtraction
* Multiplication

**E2.**

* All arithmetic operations
* Complex calculation
* Equation solving

### User Stories:

**User Story**1.

**Description:**

* As an accountant.
* I want to add 2 numbers.
* I want to see the result with less time.

**Test Case:**

* Given 2 numbers 3 and 5.
* When I add them.
* Result should be 8 with no time.

**User Story2.**

**Description:**

* As a Student.
* I want to reset the calculator.
* So that I can start fresh.

**Test Case1:**

* Given I am in the middle of an operation.
* When I press AC (clear everything) key.
* Then the operation should be cancelled.
* The display should show 0 show that it will be ready for next operation.

**Test Case2:**

* Given that the display status bar on the display shows ‘M’.
* When I press AC key.
* Then the memory should get cleared and it should not display ‘M’.
* The display should show ‘0’ for further operations