Learning Report – Module Name SDLC

GLOBAL ENGINEERING ACADEMY

Genesis

L&T Technology Services

Document History

Ver. Rel. No.	Release Date	Prepared. By	Reviewed By	Approved By	Remarks/Revision Details
99002675	22-09-2020	Afrid Shariff	Sachin C mirji(99002673) Prasanna Kumar S(99002671)		
99002675	23-09-2020	Afrid Shariff	Sachin C mirji(99002673) Prasanna Kumar S(99002671)		
99002675	24-09-2020	Afrid Shariff	Sachin C mirji(99002673) Prasanna Kumar S(99002671)		
99002675	25-09-2020	Afrid Shariff	Sachin C mirji(99002673) Prasanna Kumar S(99002671)		
99002675	26-09-2020	Afrid Shariff	Sachin C mirji(99002673) Prasanna Kumar S(99002671)		

Table of Contents

List of Figures

List of tables

Activity 1– System/Software Development

IOT based wearable smart health monitoring system

Section 1:

1.**1 Definition:** Health monitoring system is an effective way to review health condition of any condition of any individual. It helps to provide monitoring anytime anywhere. Health monitoring is a useful research area where basic routine health parameters can be reviewed anytime by any individual.

Research:

1.2.1 Aging:

Enhanced and intelligent health-care system is a symbol of developed and prosperous nation. Internet of things(IOT) has abounded the digital health-care system by providing the remote monitoring the patients' health condition and allowing doctors to have access to that information.

The product being developed are capable of monitoring pulse rate and body temperature at a time and requires some volunteer activity [1], however some of the products being used at the hospitals are capable of monitoring and generating reports on multiple parameters, which are then used by medical experts to assist the patient.

1.2.2 Costing:

Health system used at the hospitals are very cost effective and space consuming.

More over the medical staff are needed at every level from interfacing sensor to capturing the parameters.

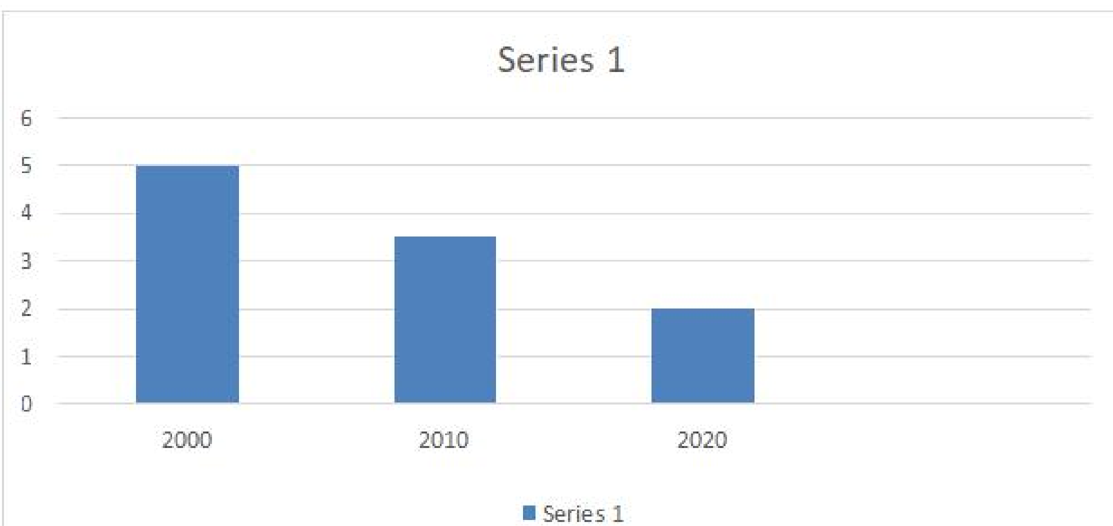


Figure 1 Cost comparison of product at different periods.

1.3 Definition of the project

The sensors which are currently being used requires external stimulus. Since the Health monitoring system used at hospitals are capable of monitoring many parameters the cost of deployment is very high.

The Simple BP sensor may cost around Rs.1500 and requires the external stimulus to be applied. Making this data to be available remotely requires additional interfaces.

There for developing a system with a pulse, blood pressure and temperature sensor along with the necessary interfaces is proposed.

1.4 SWOT analysis:

A basic SWOT analysis diagram is most commonly used to assess a product, of four criteria: strengths, weaknesses, opportunities, and threats.

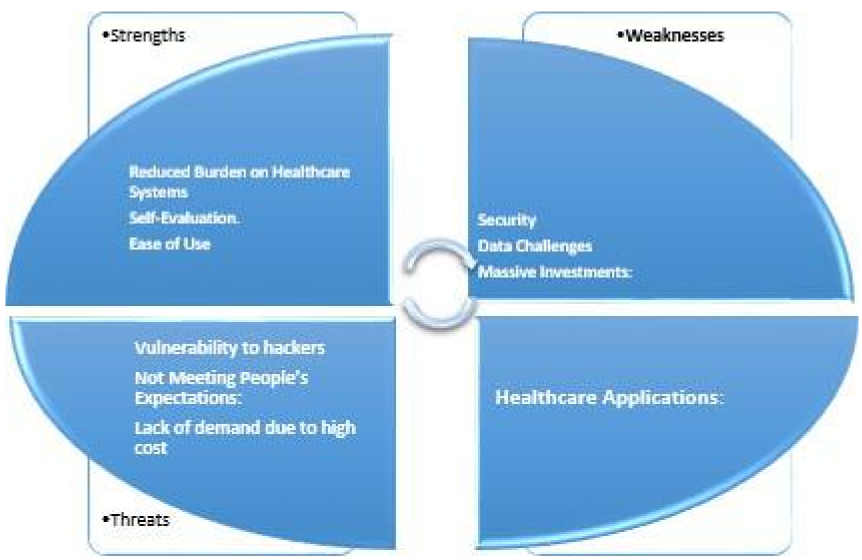


Figure 2 SWOT Analysis.

Strengths

- Reduced Burden on Healthcare Systems
- Patients can get care without going into the office, since data are collected remotely and a qualified provider analyses and responds to the data digitally, such as via email or text.
- Self-Evaluation.
- Ease of Use

Weaknesses

- Security: The most talked about drawback of connecting devices is their security and how it can be compromised by a small group of hackers. Recent activities of hackers trying to gain control over smart fridges is not doing any good to the reputation of IoT. However, there are attempts being made to revitalise security of such devices and to establish a common standard for the same.

- Data Challenges: Every year, we produce data in Exabyte's. This data needs to be stored and analysed for obtaining information about certain parameters. When all devices are connected, the amount of data collected will increase manifold. Collection, analysis and storage of all that data is an arduous task and we need better infrastructure to manage the avalanche of data headed our way.

- Massive Investments: Companies wishing to become early movers in the IoT market have to invest a lot of money to make connected devices. Apart from the production costs, there is a huge cost attached to the Research and Development of the products as well. This high cost might intimidate new market entrants. Companies need to stay poised to reap the benefits of such investments over time.

Opportunities

- Healthcare Applications: Paradigm-shifting to the field of personal healthcare is the agenda that is leading the revolution of connected devices. There are several opportunities for developers to innovate and make solutions to make our lives easier. Recent development of the Health Kit and Research Kit by Apple is just a step forward in the direction of improving healthcare. With so many ideas perking up every now and then and the state of current technology, anything is possible. As potential customers, this is a win-win situation for us.

Threats

- Vulnerability to hackers: When we think about hackers, we imagine really intelligent individuals working tirelessly on their Alien wares to bring down a website. Now, with connected devices in the picture, these hackers will be able to control your smart bulbs, garage, watches and even clothes! This open invitation to hackers to try to control every device around is a serious threat for IoT and it stands in the way of users shifting to connected devices.

- Not Meeting People's Expectations: A classic example to bring home this point can be the console game, Watchdogs. The game was previewed at a conference when it was in its early stages of development. People pinned hopes high expecting an open-world hacking game. But, what was delivered did not live up to the expectations, making it an average product after all. IoT here has reached the peak of it hype. People have realistic and over-the-top hopes from IoT. These exaggerated expectations are a threat to IoT if the products fail to live up to the user expectations.

- Lack of demand due to high cost: Let's face it, smart watches, glasses or even bulbs are not cheap. A pack of 3 smart bulbs is almost three times the cost of regular ones. It is great for us that companies are developing connected devices, but they will be of no use if the intended target audience is not able to afford them. Large selling prices is a very big threat looming over IoT and its growth.

1.5 Requirements:

To design a remote real-time monitoring system for multiple physiological parameters based on IOT, including a heart rate, blood Pressure and temperature. The proposed system should meet fallowing requirements

1.5.1 High Level:

such as sensing, processing, displaying, and real-time transmission and should have the capability to generate an auto alarm based on the analysis threshold values of multiple monitoring parameters. The system should also meet auxiliary requirements such as compatibility, comfort, low power consumption and cost, and smallsize.

Hardware components used:

Arduino uno

ESP8266.

Heart Rate Pulse Sensor.

LM35.

Blood pressure sensor (should be able to generate the serial output)

LCD Display-16x2.

Software requirements:

Arduino IDE.

Thinks Speak.

1.5.2 Low level:

To meet mentioned high level requirement, the fallowing modules can be used

BLOOD PRESURE SENSOR: Blood Pressure & Pulse reading are shown on display with serial out for further processing. Shows Systolic, Diastolic and Pulse Readings. Compact design fits over your wrist like a watch. Easy to use wrist style eliminates pumping.

Figure 3 Blood pressure sensor module.

Specification

Working Voltage: +5V, 200mA regulated.

Output Format: Serial Data at 9600 baud rate(8 bits data, No parity, 1 stop bits). Outputs three parameters in ASCII.

Sensing unit wire length is 2 meters.

Sensor Pin outs

TX-OUT = Transmit output. Output serial data of 3V logic level, usually connected to RXD pin of microcontrollers/RS232/USB-UART.

+5V = Regulated 5V supply input.

GND = Board Common Ground.

Section 2

2.1Design:

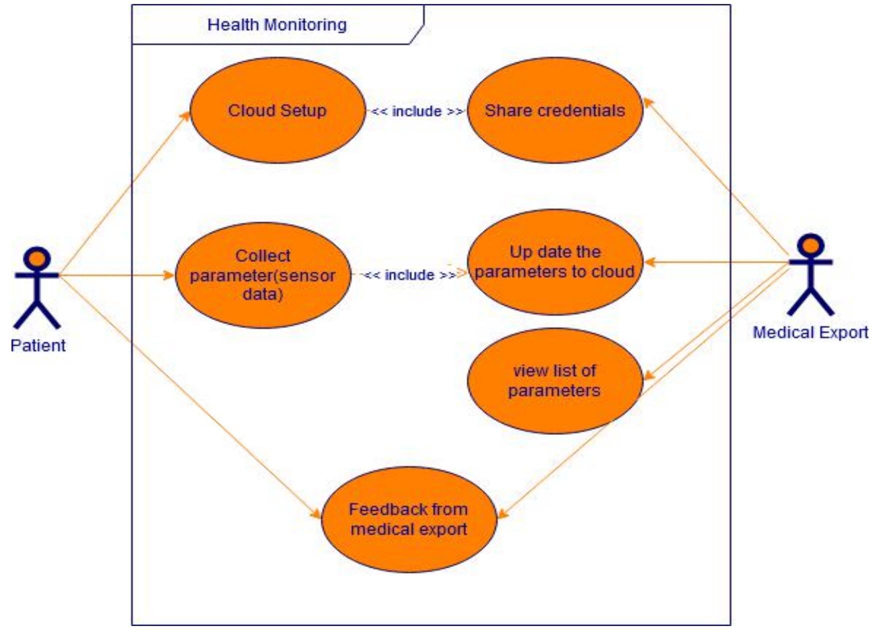


Figure 4 use case diagram high level

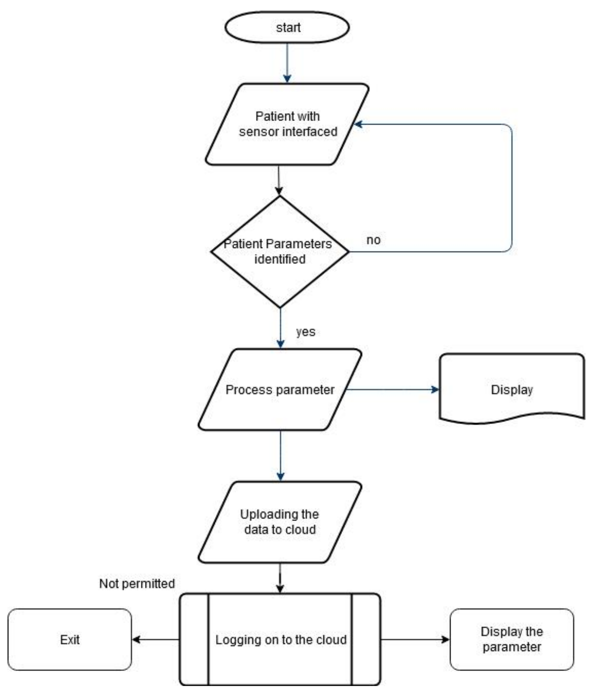


Figure 5 Activity diagram high level

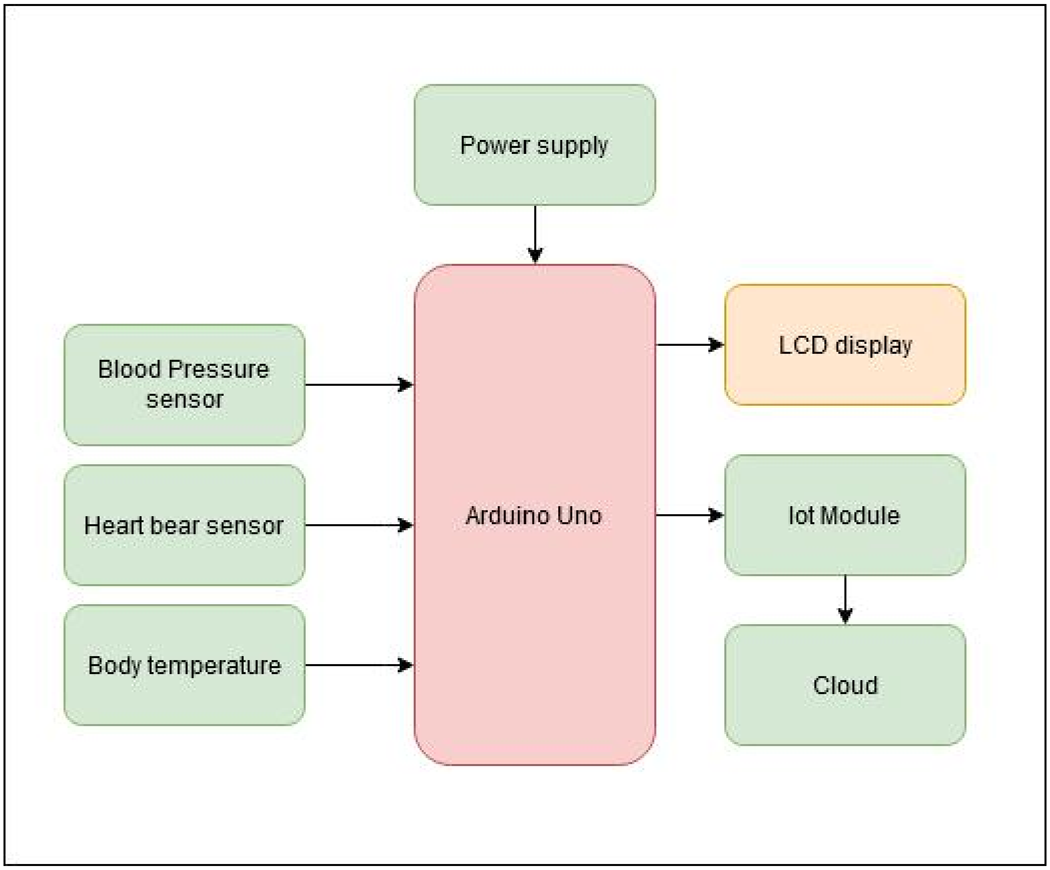


Figure 6 Component Diagram High Level

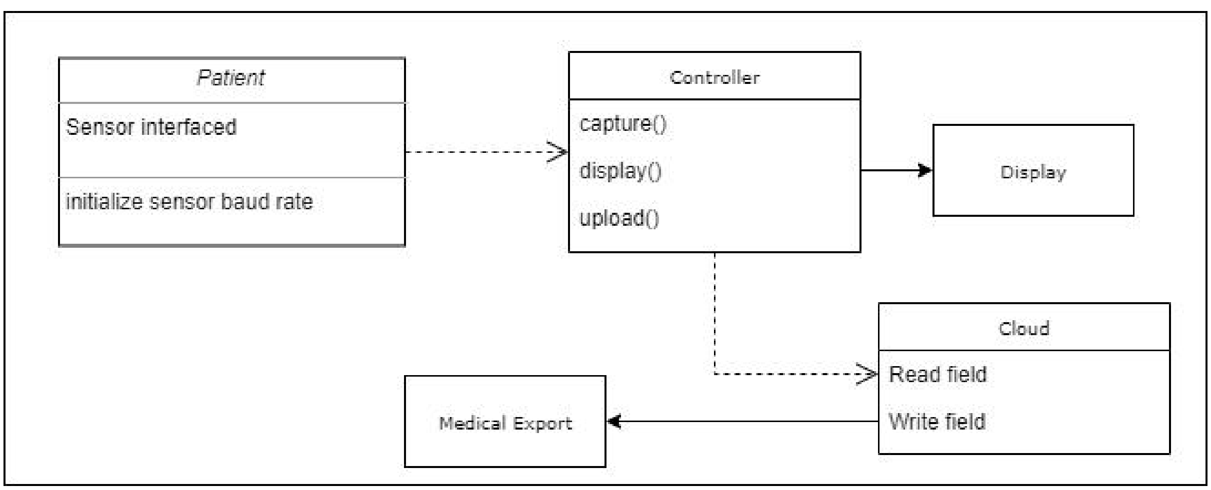


Figure 7 Class Diagram High Level

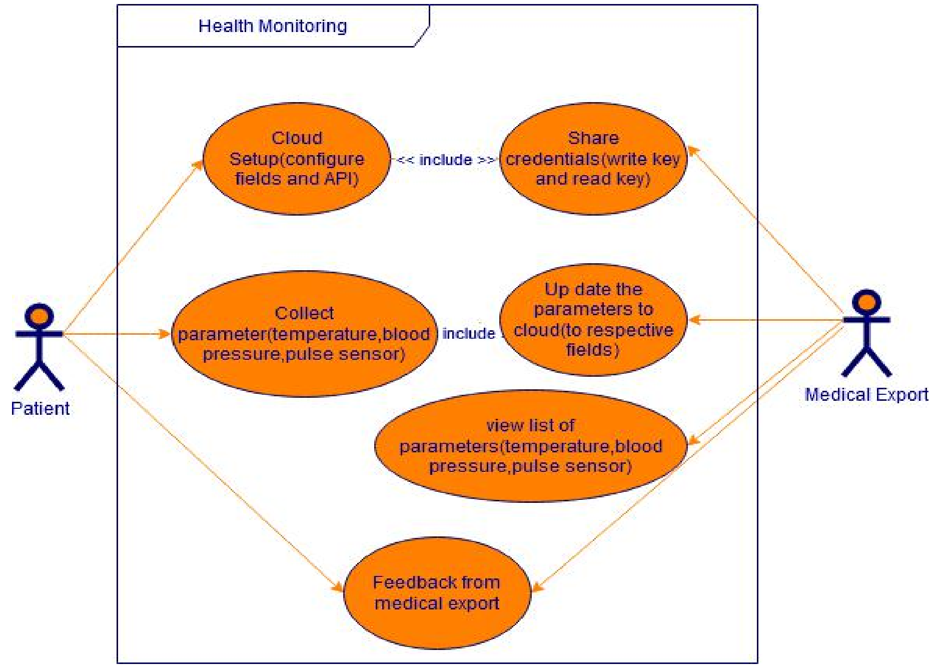


Figure 8 use case diagram low level

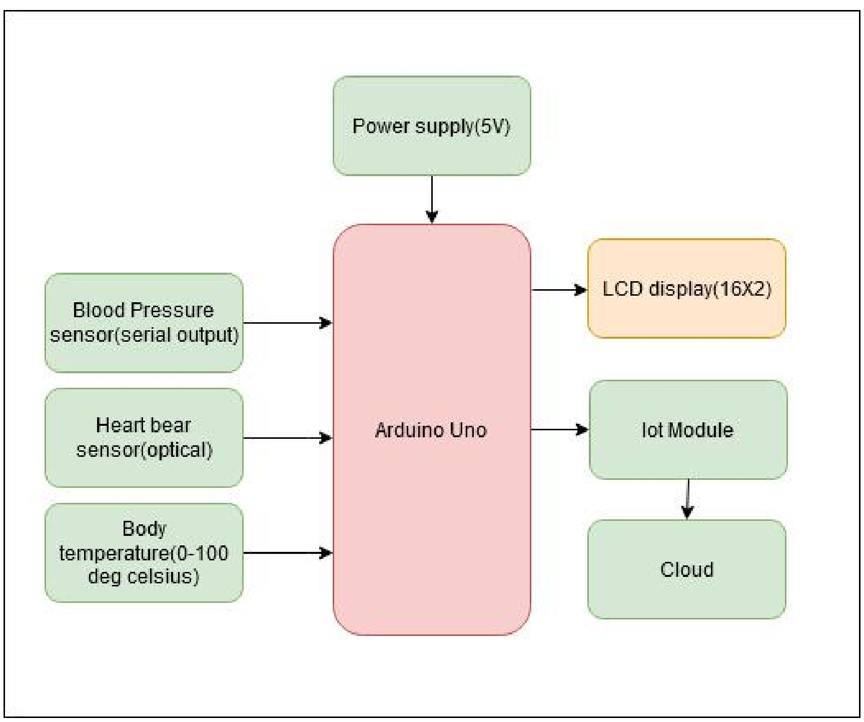


Figure 9 component Diagram Low Level

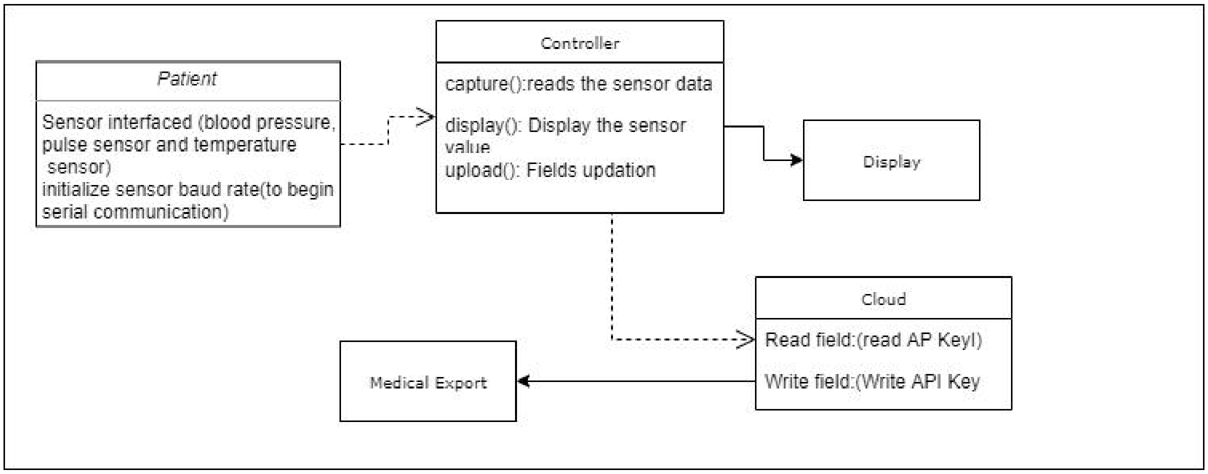


Figure 10 Class Diagram High Level

Section 3:

3.1 Test Plan

Requirement Based Test Plan.

- Test plan 1: Supply voltage of 7v to 12 v to power up the Arduino Uno.
- Test plan 2: Baud Rate to be initialized 9600.
- Test plan 3: Serial pins (Rx & TX) of Arduino must be connected the TX &Rx pins of blood Pressure sensor.

Scenario Based Test Plan.

- Test plan 4: Configuring the digital pins as Rx & TX.
- Test plan 5: Exchanging Rx and TX pins.

Boundary Condition based test plan.

- Test plan 6: Selecting Baud rate above 9600.
- Test plan 7: Selecting Baud rate below 9600.
- Test Plan 8: Supply voltage below 5v.

Activity 2 – Agile Aspects

2.1. High level Requirement:

ID	Description
HL-01	Supply voltage for Arduino (5V)
HL-02	Output Format: Serial Data
Hl-03	Use serial Communication pins available
HL-04	Digital pins for communication(configure)
HL-05	Sensing unit wire length is 2 meters.
HL-06	Inbuilt supply for sensor in case of standalone unit
HL-07	Memory to store the data

Table 1 High level Requirement

2.2. Low level Requirement:

ID	Description
LL-01	Supply voltage for Arduino between +7v to +16v and Blood pressure sensor requires +5v, 200mA regulated supply
LL-02	Serial data at 9600 baud rate (8 bits' data, No parity, 1 stop bits). Outputs three parameters in ASCII. • Systolic • Diastolic • Pulse
LL-03	Serial pins(digital pin 0 & digital pin 1) (Rx & TX) of Arduino must be connected the TX &Rx pins of blood Pressure sensor.
LL-04	Configure the Digital pins(2 to 13) of Arduino to use Rx & Tx
LL-05	Sensing unit wire length is 2 meters with serial data converter.

LL-06	Use a battery of +5v output.
LL-07	Memory should be able to store values of 60 trails.

Table 2Low level Requirement:

2.3. Test plan

Test ID	Description	Prerequisite	Expected input	Expected output	Actual output	Remarks
HH-01 LL-01	Supply voltage for Arduino between +7v to +16v and Blood pressure sensor requires +5 v, 200mA regulated supply	Input voltage 230v stepdown to 5V DC or Plug the board into a USB port on your computer	Input:5V DC	Check that the green LED power indicator is ON located near the reset switch.	Green LED power indicator is ON located near the reset switch.	Pass
HH-01 LL-01	Supply voltage for Arduino between +7v to +16v and Blood pressure sensor requires +5 v, 200mA regulated supply	Input voltage 230v stepdown to 5V DC or Plug the board into a USB port on your computer	Input: Below 4.9v DC	Check that the green LED power indicator is OFF located near the reset switch.	Green LED power indicator is ON located near the reset switch.	Fail

HH-02 LL-02	Serial data at 9600 baud rate (8 bits' data, No parity, 1 stop bits). Outputs three parameters in ASCII. • Systolic • Diastolic • Pulse	Initialize the baud rate to 9600	Input : baud rate 115200	Check whether able to read the data coming at serial pins of Arduino	Garbage data obtained	Fail
HH-03 LL-03	Checking Serial pins(digital pin 0 & digital pin 1) (Rx & TX)	Serial pins(digital pin 0 & digital pin 1) (Rx & TX) of Arduino must be connected the TX &Rx pins of blood Pressure sensor.	Systolic, Diastolic and Pulse	Systolic, Diastolic and Pulse output at LCD	Systolic, Diastolic and Pulse output at LCD observed	Pass
HL-04 LL-04	configuring Digital pins for Rx and TX	Configure the Digital pins(2 to 13) of Arduino to use Rx & Tx	Systolic, Diastolic and Pulse	Systolic, Diastolic and Pulse output at LCD	Systolic, Diastolic and Pulse output at LCD observed	Pass
HL-05 LL-05	External supply check	Connect a battery of +5v output.	Input:5V DC	Check that the green LED power indicator is ON located near the reset switch.	Green LED power indicator is ON located near the reset switch.	Pass
HL-05 LL-05	External supply check	Connect a battery of +4v output.	Input:4V DC	Check that the green LED power indicator is ON located	Green LED power indicator is OFF located near the reset switch.	Fail

				near the reset switch.		
HL-07 LL-07	Memory	Make sure memory installed in BP sensor	Do trails for 60 patients	Check for 60 Patients data	60 patient data identified	
HL-07 LL-07	Memory	Make sure memory installed in BP sensor installed in BP sensor	Do trails for 30 patients	Check for 60 Patients data	30 patient data identified	

Table 3 Test Plan

2.4 User stories:

1. **Description:**
    a. I want to send my blood pressure, pulse rate and body temperature to my doctor.

    Test case:
    a. Given the module you can interface and monitor the parameters such as blood pressure, pulse rate and body temperature, as well as uploading the value the cloud
    b. Doctor can view the parameters by accessing the cloud and gives he necessary feedback/alert message to the patient.
2. **Description:**
    a. I want to store blood pressure of 60 people.

    Test case:
    a. Given the module you can interface and record the parameters of 60 patients and upload the same to cloud.
3. **Description:**
    a. I want to know ma blood pressure remotely and want upload to cloud.

    Test case:
    a. Given the module you can interface and record the parameters.
    b. Upload he values using API keys.

Activity 3 –System Software development for geneses

- Badges

  Cpp checkLink Link

  Code quality

  Unity Testing Link

  CI passing Link

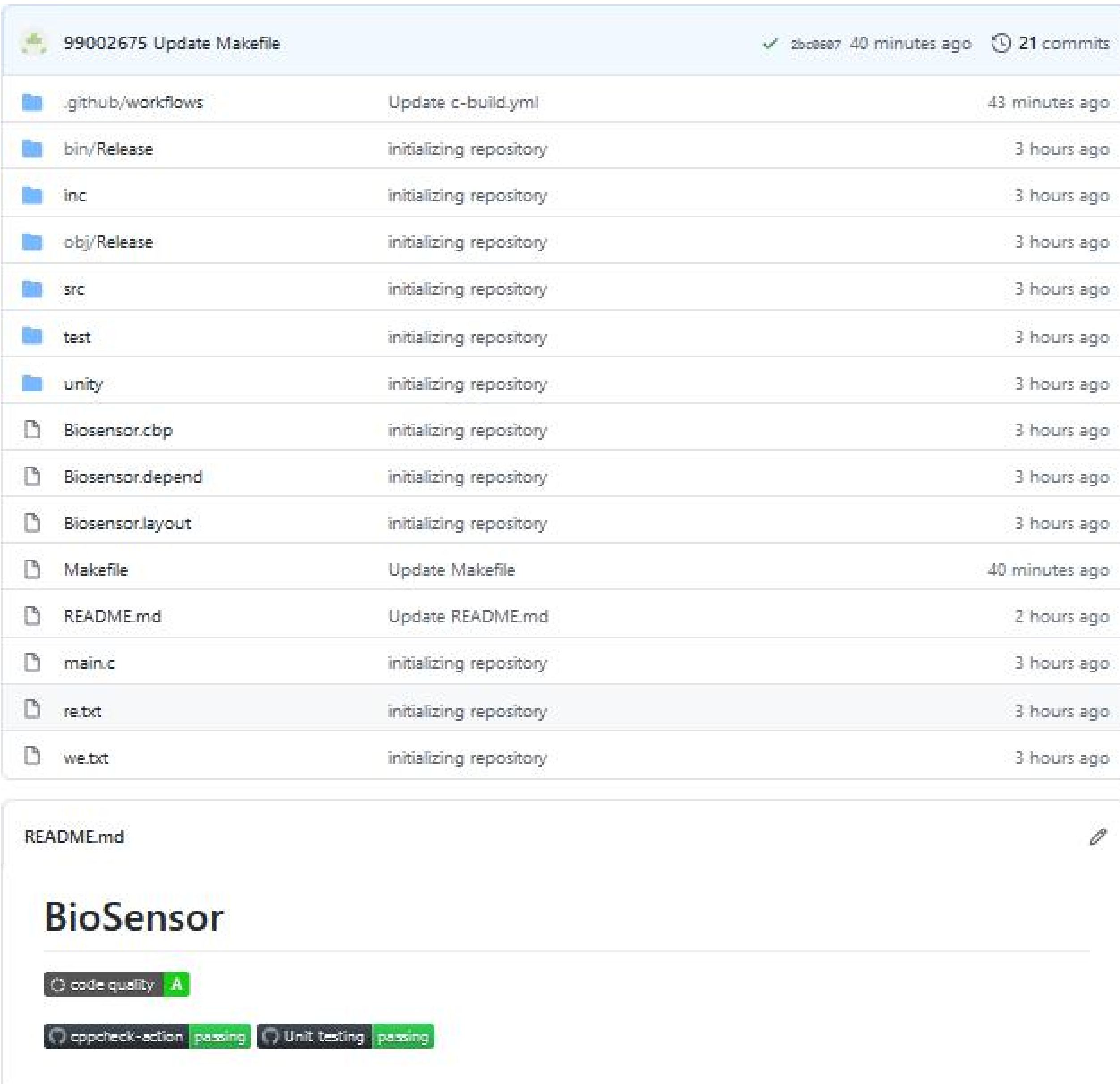


Figure 11 Git Hub Badges

- Commits [Link]()

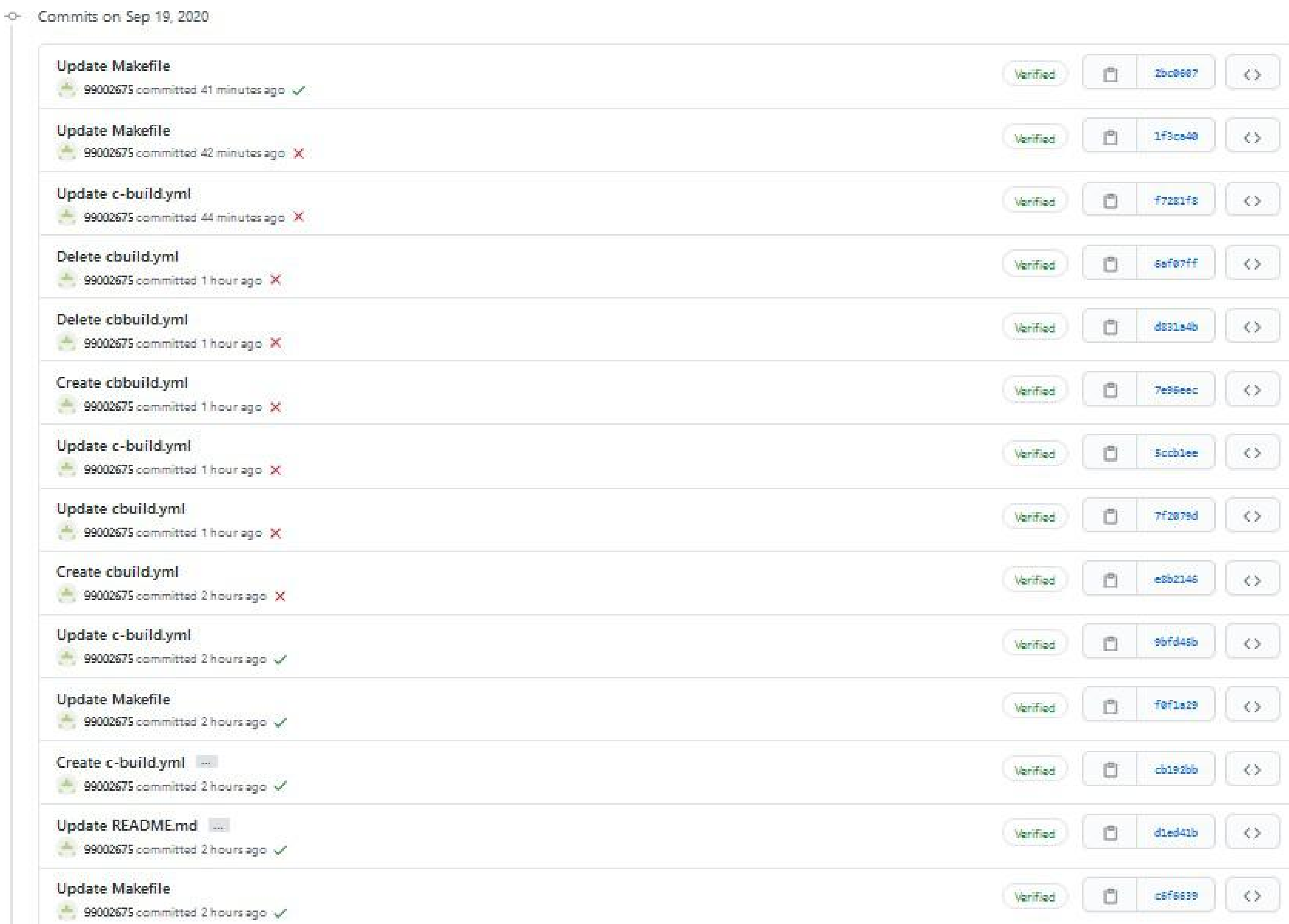
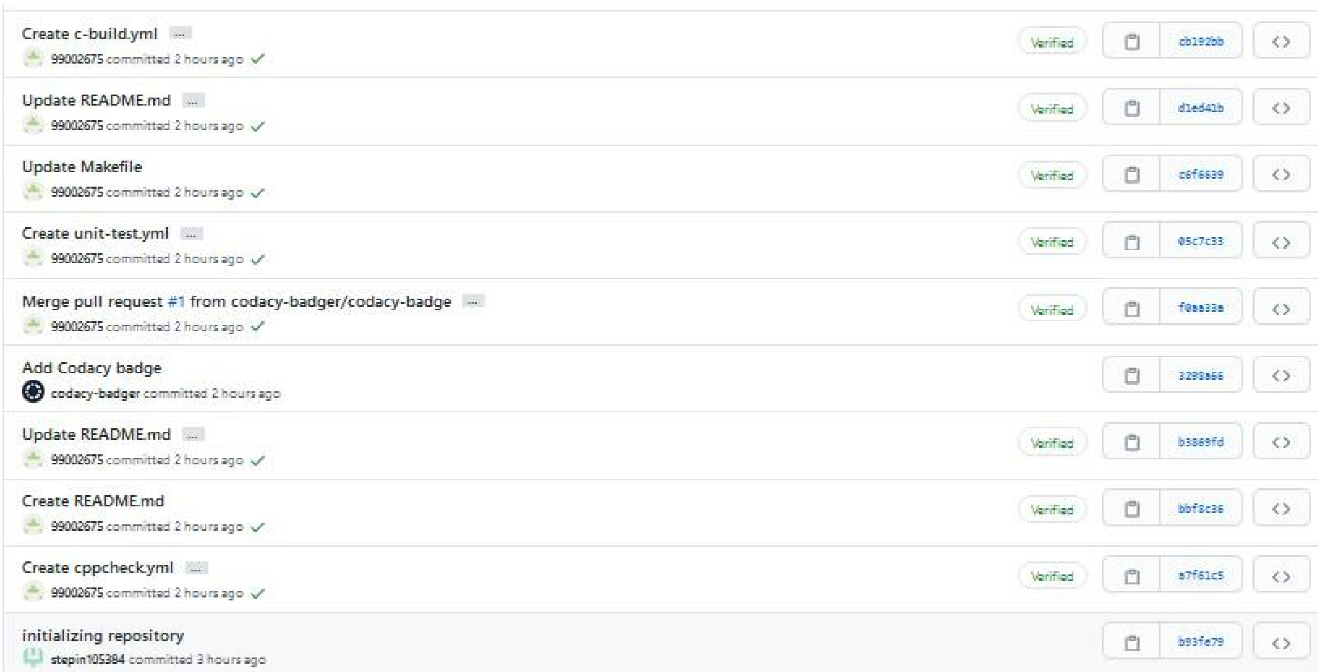


Figure 12 Commit History



- Build and Make file [Link]

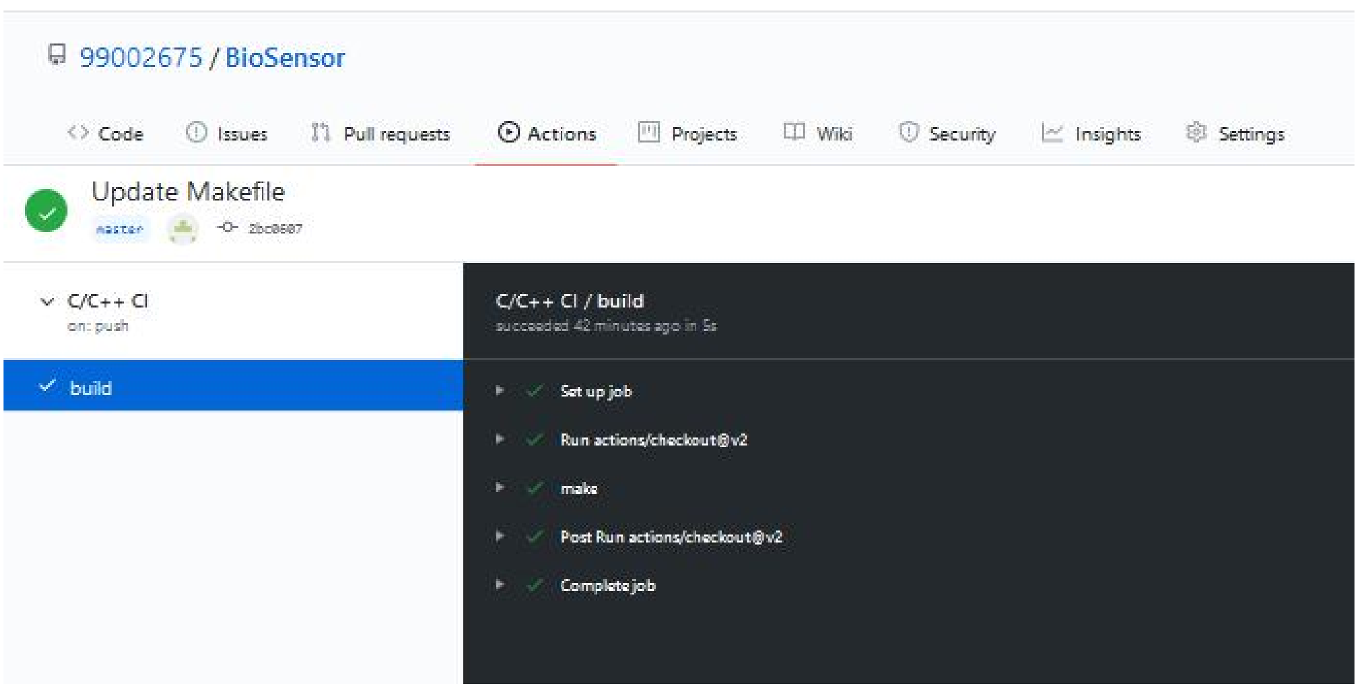


Figure 13 Build and make file status

- Issue Raising [Link]

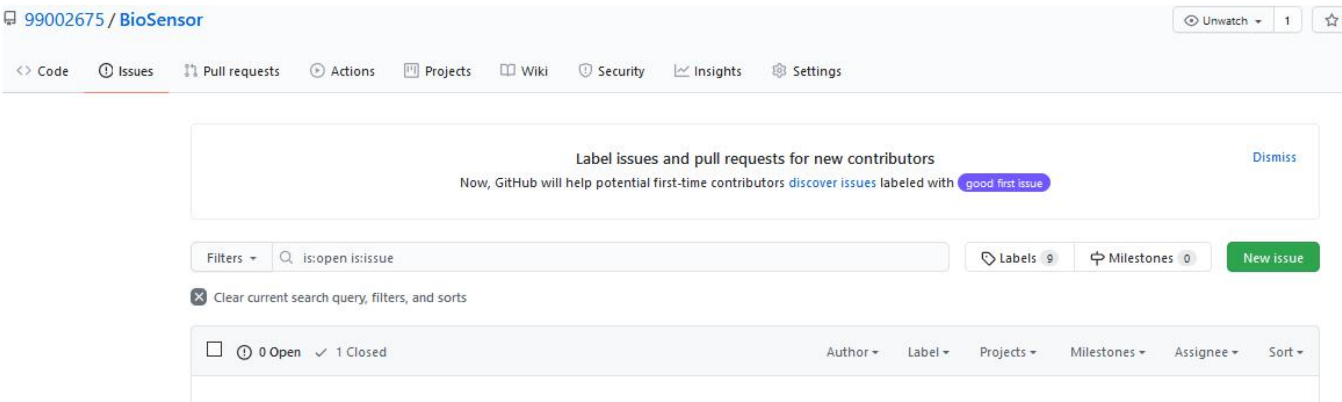


Figure 14 Issue Raising

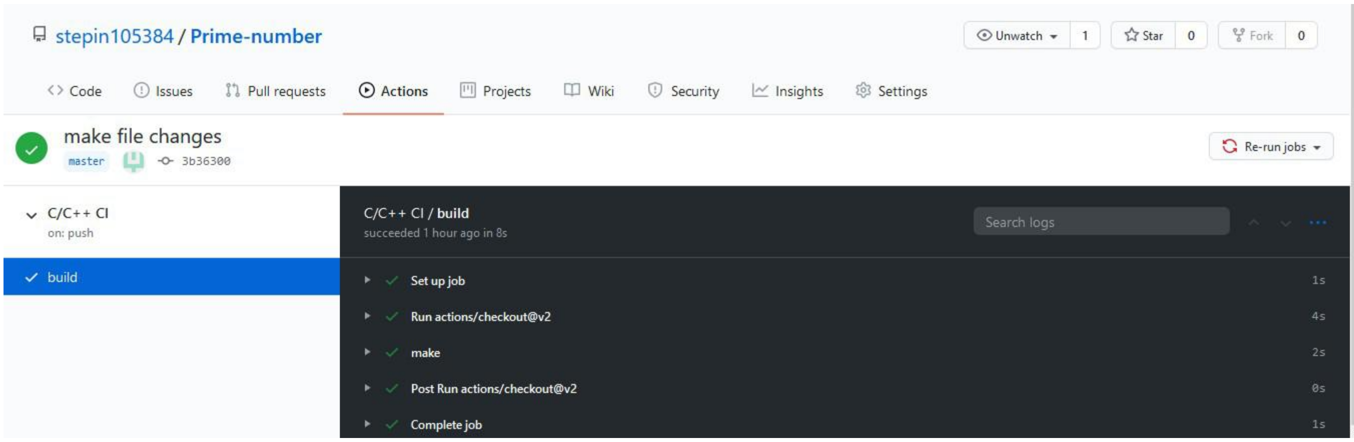
- Make File Code:

```
SRC = unity/unity.c\
src/biosensor.c\
test/test_biosensor.c\
main.c

INC = -Iunity\
-Iinc\
-Itest

PROJECT_NAME = BIOSENSOR.out

$(PROJECT_NAME): $(SRC)
    gcc $(SRC) $(INC) -o $(PROJECT_NAME)

run:$(PROJECT_NAME)
    ./${PROJECT_NAME}
doc:
    make -C documentation

clean:
    rm -rf $(PROJECT_NAME) documentation/html
```

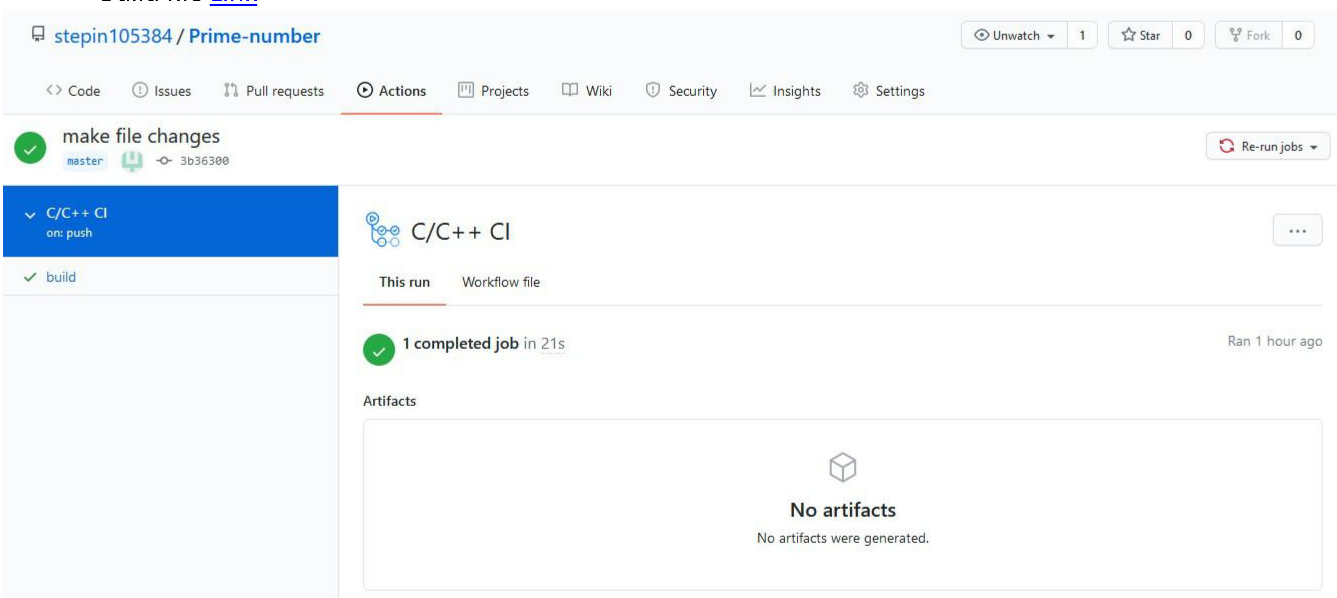
Git Repository link: [Link]

Appendix: CI Workflow for C Programming

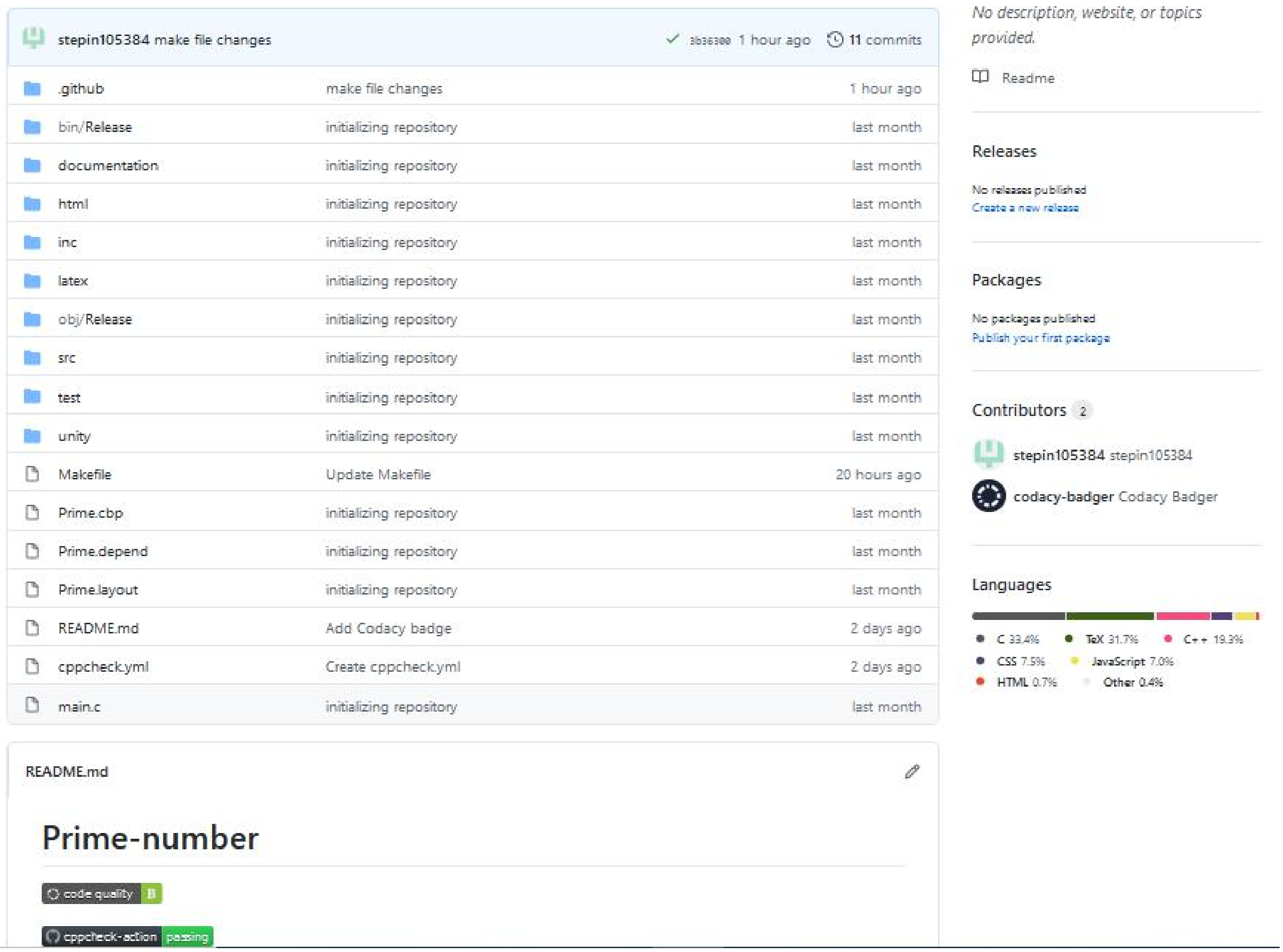
- Make file Link



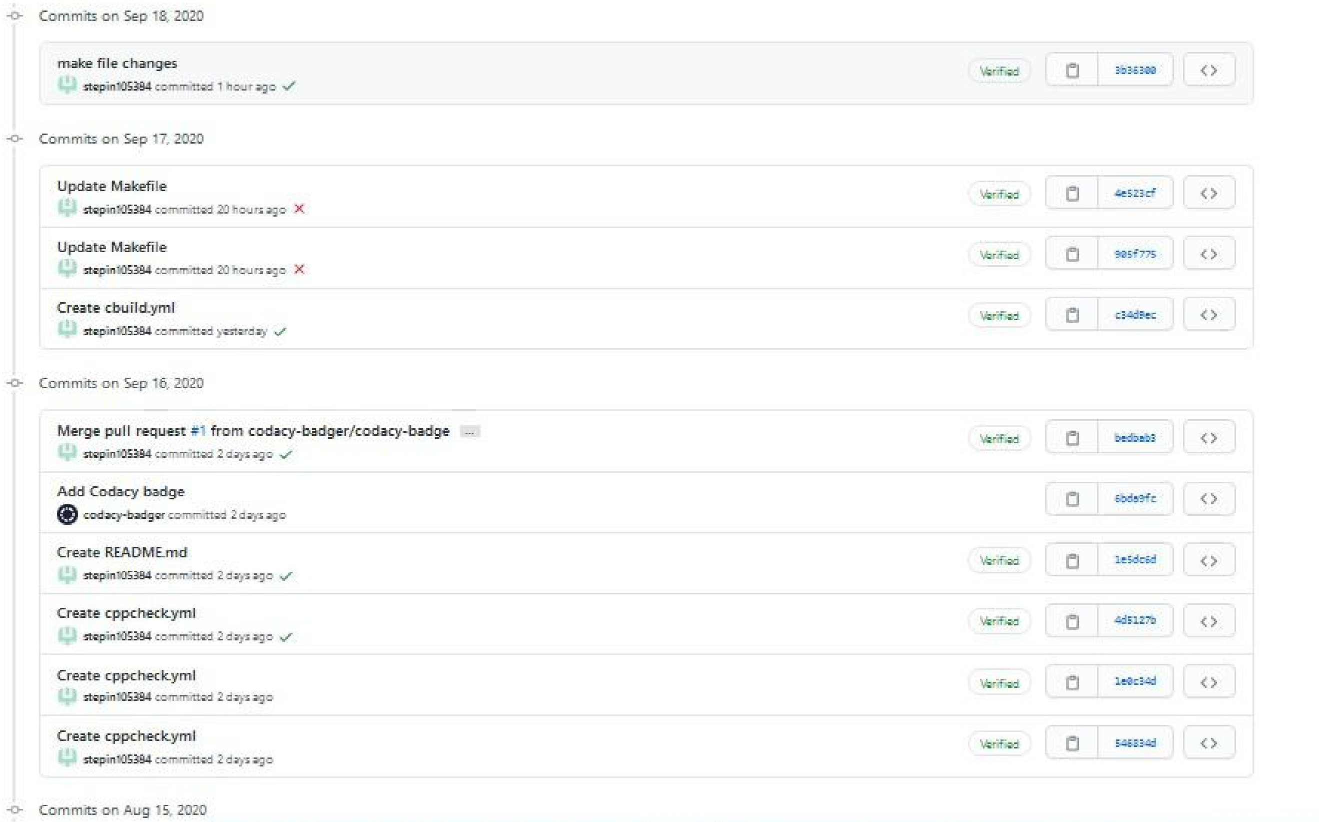
- Build file Link



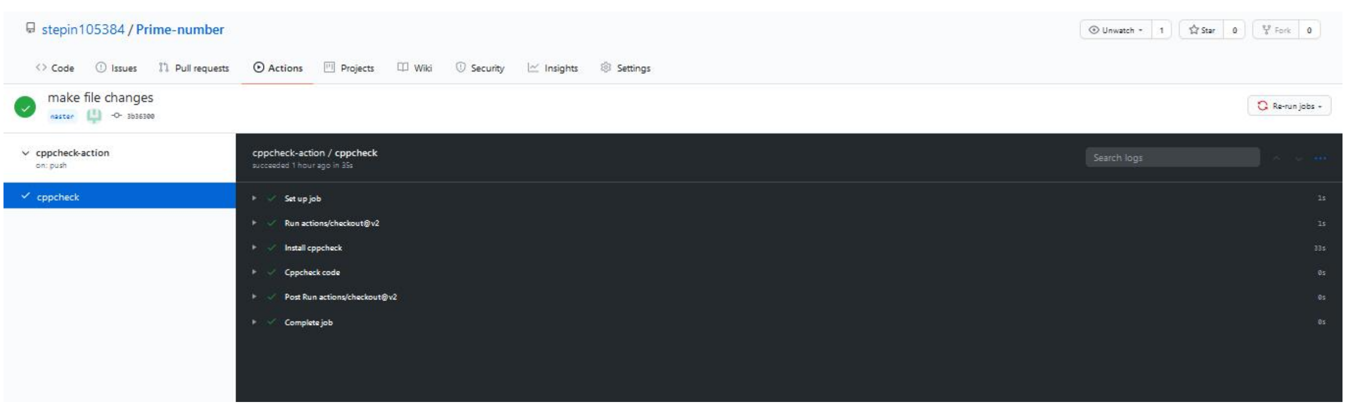
- Cpp check [Link](#) and code Quality [Link](#)



- Commits



- Cpp check [Link]



- Make File code:

```
SRC = unity/unity.c\
src/prime.c\
test/test_prime.c\
main.c
INC = -Iunity\
-Iinc\
-Itest
PROJECT_NAME = PRIME.out
$(PROJECT_NAME): $(SRC)
	gcc $(SRC) $(INC) -o $(PROJECT_NAME)
run:$(PROJECT_NAME)
	./${PROJECT_NAME}
doc:
	make -C documentation
clean:
	rm -rf $(PROJECT_NAME) documentation/html
```

Git Link: https://github.com/stepin105384/Prime-number

References:

[1] https://circuitdigest.com/microcontroller-projects/iot-based-patient-monitoring-system-using-esp8266-and-arduino

[2] https://www.sunrom.com/p/blood-pressure-sensor-serial-output