./

Learning Report – Applied SDLC and Software Testing



Sourab John Jacob

99002447

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **Approved By** | **Remarks/Revision Details** |
| 1.1 | 17/9/2020 | Sourab John Jacob |  |  |  |
| 1.2 | 18/9/2020 | Sourab John Jacob |  |  |  |
| 1.3 | 19/9/2020 | Sourab John Jacob |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

# 

Contents

[Table of Figures 4](#_Toc51416368)

[Checklist 4](#_Toc51416369)

[Activity 1 – System/Software Development 5](#_Toc51416370)

[1. Introduction 6](#_Toc51416371)

[2. Product Definition 7](#_Toc51416372)

[3. SWOT Analysis 7](#_Toc51416373)

[4.Requirements Gathering 7](#_Toc51416374)

[4.1 High Level Requirements: 7](#_Toc51416375)

[4.2 Low Level Requirements: 7](#_Toc51416376)

[5. Design Models 8](#_Toc51416377)

[5.1Structural Diagrams 8](#_Toc51416378)

[5.2Behavioral Diagrams 10](#_Toc51416379)

[6.Test Plan 12](#_Toc51416380)

[7. References 12](#_Toc51416381)

[Activity 2 –Agile Concepts 13](#_Toc51416382)

[Theme: 13](#_Toc51416383)

[Epic: 13](#_Toc51416384)

[User Stories: 13](#_Toc51416385)

[Activity 3 –GitHub Final Submission 14](#_Toc51416386)

[1. Introduction 15](#_Toc51416387)

[2. Requirements Gathering 15](#_Toc51416388)

[2.1 High Level Requirements 15](#_Toc51416389)

[2.2 Low Level Requirements 16](#_Toc51416390)

[3. UML 16](#_Toc51416391)

[3.1 Use case Diagram 16](#_Toc51416392)

[3.2 Class Diagram 17](#_Toc51416393)

[3.3 Sequence Diagram 18](#_Toc51416394)

[4. Test Plan 19](#_Toc51416395)

[5. GitHub Repository 19](#_Toc51416396)

[6. References 19](#_Toc51416397)

[Appendix –CI Workflow for C Programming (StepIn) 20](#_Toc51416398)

[1.Git 21](#_Toc51416399)

[2.Make 22](#_Toc51416400)

[3.Build 22](#_Toc51416401)

[4.Code Quality 23](#_Toc51416402)

**Table of Figures**

[Figure 1 Cost Grading Graph 7](#_Toc51411515)

[Figure 2 Component Diagram 8](#_Toc51411516)

[Figure 3 Class Diagram 9](#_Toc51411517)

[Figure 4 Flow Chart 10](#_Toc51411518)

[Figure 5 Sequence Diagram 11](#_Toc51411519)

# Checklist

* Installation of SW on Phone and Desktop
* Registering LTTS credentials
* Reacting to Polls and communicating on Teams and Yammer
* Using StepIn credentials for GitHub for Activity 2
* Creating New GitHub account using LTTS credentials
* Creating new repository for final submission
* Building Requirements and UML diagrams
* Creating Test Plan
* Coding the program
* Testing it against unit test cases
* Finalizing and Evolving all documents

Activity 1 – System/Software Development

**Evolution of Smart Phone Battery Technology**

**1.** Introduction

The first working practical batteries were the Daniel cell, Birds cell, Porous pot cell etc. There has been a lot of innovation in this field to improve the efficiency and charging speeds. Most smart phones today use lithium ion batteries and we know how dangerous these could be at times. We can recall the recent case of Samsung galaxy note 7 phones where there were numerous cases reported about blasts due to the faulty batteries. This leads to major losses to such multibillion dollar companies.

There has been a shift from lithium ion to li-poly and even li-phosphorous batteries which have improved the efficiency. But the major problem still that exists is that Lithium is not a naturally occurring metal. Lithium-ion batteries have a rather volatile liquid electrolyte porous material layer sandwiched between the anode and cathode layers.

The latest innovations in batteries are Ryden dual carbon battery, Sodium-ion batteries, Upp hydrogen fuel cell charger. Scientist in Japan have developed such batteries which have up to 20X faster charging speeds. Not only will it last longer and charge faster than lithium but it can be made using the same factories where lithium batteries are built. The batteries use carbon materials which mean they are more sustainable and environmentally friendly than current alternatives

Scientists in Japan are working on new types of batteries that don't need lithium like your smartphone battery. These new batteries will use sodium, one of the most common materials on the planet rather than rare lithium – and they'll be up to seven times more efficient than conventional batteries.

COST on Y axis GRADE on X axis

Figure Cost Grading Graph

2. Product Definition

**A Smartphone battery system that has the ease of manufacturing as lithium ion batteries but with the benefits of graphene and other newer battery systems.**

Lithium-ion batteries are compact and readily available to be used and manufactured, the transition to newer technologies at scale will be a difficult task, How can the newer proof of concepts be brought to the masses in the next few years?

3. SWOT Analysis

**Strengths**-Increasing Usage of li-ion batteries in Industrial applications. Growth in Mobile and Computing Technologies to boost Li-ion Battery usage in Consumer Electronics

**Weakness**- Safety Concerns, Generation of e-waste as new phones are being released by the minute, High cost of li-ion batteries.

**Opportunity**- Compact structure of these batteries enable it to be used in different appliances.

**Threats**-The industry is weary of Li-ion batteries due to a history of fire incidents.

Newer greener alternatives like graphene batteries are being developed. Graphene batteries have many benefits but the one shortcoming that’s holding its mass-adoption in our devices is mass production and the costs involved in the same.

4.Requirements Gathering

# **4.1 High Level Requirements**:

H1. Efficient Battery System that can power mobile phones for multiple days and support fast charging.

# 4.2 Low Level Requirements:

* Fast Charging Speeds using very high-power chargers like 150W bricks to charge the battery in few minutes.
* Splitting the batteries and charging it parallelly.
* Access to graphene batteries and hydrogen cells

5. Design Models

# 5.1Structural Diagrams

Component Diagram

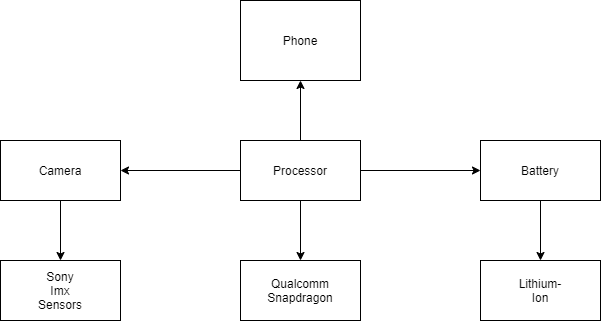


Figure Component Diagram

Class Diagram

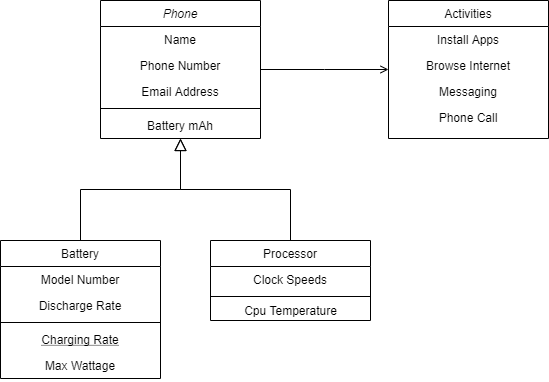


Figure Class Diagram

# 5.2Behavioral Diagrams

Flow Chart

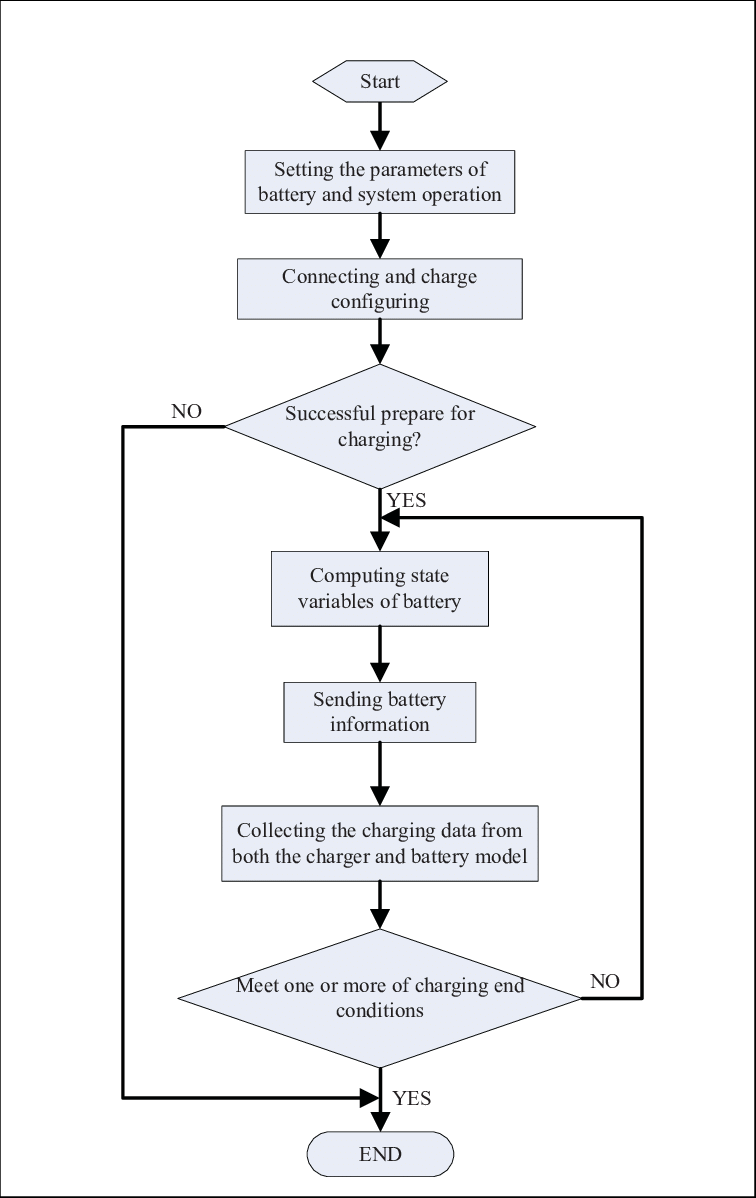


Figure Flow Chart

Sequence Diagram

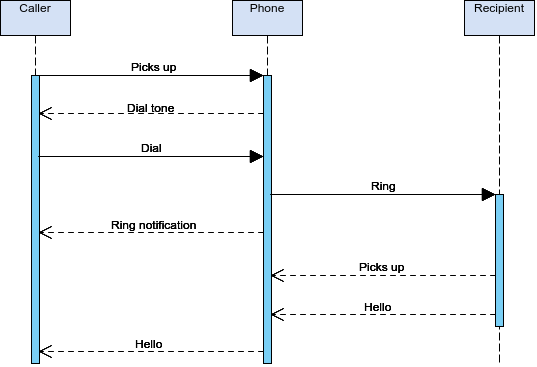


Figure Sequence Diagram

6.Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case | Precondition | Test data | Expected Output | Actual Output | Status |
| H1\_LL1 | Phone is Working | Status of Battery | When the battery is less than 10% notify and switch on the battery saver | Notification displayed or battery saver switched on | Pass |
| H1\_LL2 | Phone is Working | Status of Battery | When the battery is less than 10% notify and switch on the battery saver | Notification not displayed or battery saver not switched on | Fail |
| H1\_LL3 | Phone is Working | Status of Battery | On very less charge notify and switch off | Notified and switched the watch off | Pass |
| H1\_LL4 | Phone is Working | Status of Battery | On very less charge notify and switch off | Not Notified or not switched the watch off | Fail |

7. References

1. <https://en.wikipedia.org/wiki/History_of_the_battery>
2. <https://www.pocket-lint.com/gadgets/news/130380-future-batteries-coming-soon-charge-in-seconds-last-months-and-power-over-the-air>
3. <https://www.google.com/search?q=samsung+note+7+blast&rlz=1C1NHXL_enIN890IN891&oq=samsung+note+7+bla&aqs=chrome.0.0j69i57j0l6.7598j0j7&sourceid=chrome&ie=UTF-8>
4. <https://www.google.com/search?q=swot+analysis+for+lithium-ion+battery&rlz=1C1NHXL_enIN890IN891&tbm=isch&source=iu&ictx=1&fir=18glpg9_xQE5EM%252C4_0NwScccBy4PM%252C_&vet=1&usg=AI4_-kQannGCKFpWHylKxnKZqViYLp7fEQ&sa=X&ved=2ahUKEwjvvJ_h9ezrAhWkguYKHS2YD7EQ9QF6BAgLEEY&biw=690&bih=779#imgrc=VZnEyOASby0fbM>
5. <https://beebom.com/graphene-battery-vs-lithium-ion-battery/>

Activity 2 –Agile Concepts

Theme:

An efficient Battery System that leverages new technologies to provide fast charging and sustainable batteries in Smartphones.

Epic:

1. Battery Level Dependent automated actions

User Stories:

1. As a General User, I want the phone to last 2 days on a single charge so that I can avoid wasting charge cycles.

2. As a Gamer, I want the phone to charge in a few minutes so that I can avoid interruptions during gaming sessions.

3. As a Developer, I want the battery to be safe for the masses so that no issues are caused later.

Activity 3 –GitHub Final Submission

**Advanced Calculator Program**

**1.** Introduction

A **Software calculator** is a calculator that has been implemented as a computer program, rather than as a physical hardware device. They are among the simpler interactive software tools, and, as such, they:

* Provide operations for the user to select one at a time.
* Can be used to perform any process that consists of a sequence of steps each of which applies one of these operations.

Have no purpose other than these processes, because the operations are the sole, or at least the primary, features of the calculator, rather than being secondary features that support other functionality that is not normally known simply as calculation.

As a *calculator*, rather than a computer, they usually:

* Have a small set of relatively simple operations.
* Perform short processes that are not compute intensive.
* Do not accept large amounts of input data or produce many results.

**2.** Requirements Gathering

1) The calculator is developed using standard C and should run on all machines supporting GCC compiler

2) The program must include and display the following menu to the user: 1 Add 2 Subtract 3 Multiply 4 Divide 5 Power 6 Root 7 Factorial 8 Log 9 Log10 10 Exit

3) The program should only exit when the user enters 10

4) The program must print the result in equation form. [Ex.] <num1> + <num2> <result> = 4

5) The program must prevent the user from dividing by zero.

6) The program should display Invalid selection" if the user choses a menu option that is less than 1 or greater than 10.

7) operations will use either one or two numbers.

8) The program should prompt the user to provide numbers for the selected operation N

# 2.1 High Level Requirements

H1. Arithmetic Operations

H2. Exponential Operations

H3. Conversions

H4. Additional Functions

# 2.2 Low Level Requirements

L1. Addition

L2. Subtraction

L3. Division

L4. Multiplication

L5. Power

L6. Root

L7. Log

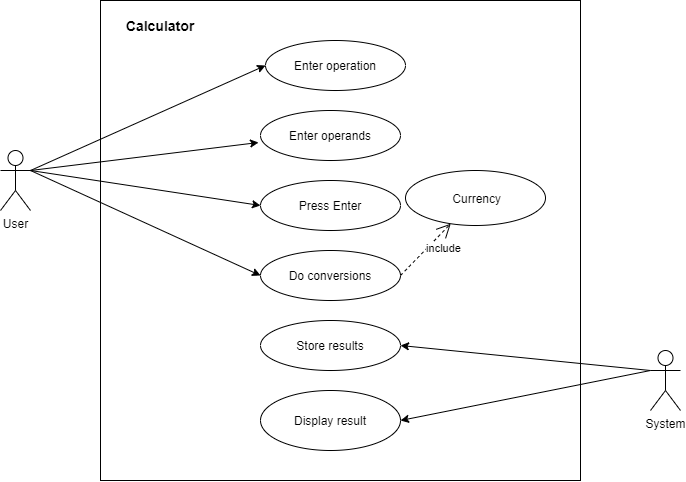
L8. Factorial

L9. Prime

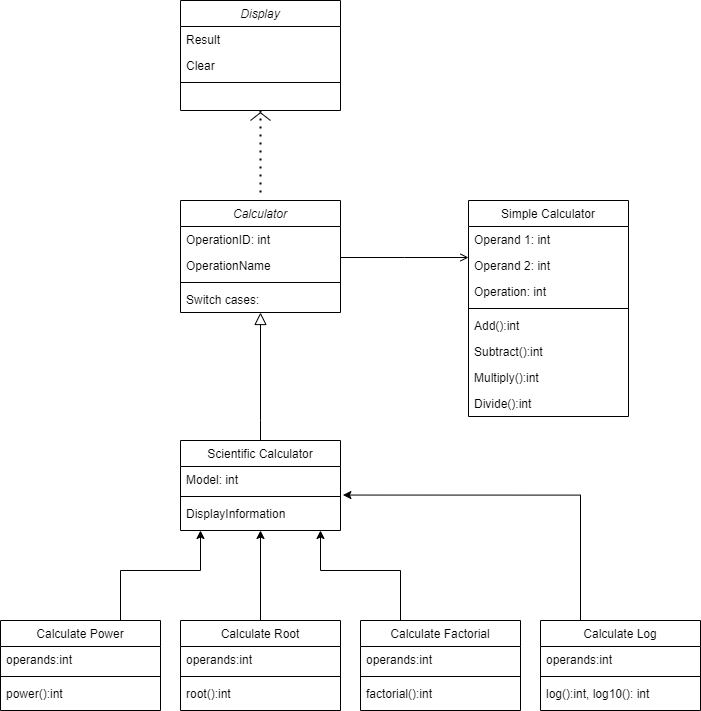
L10. Conversions

**3.** UML

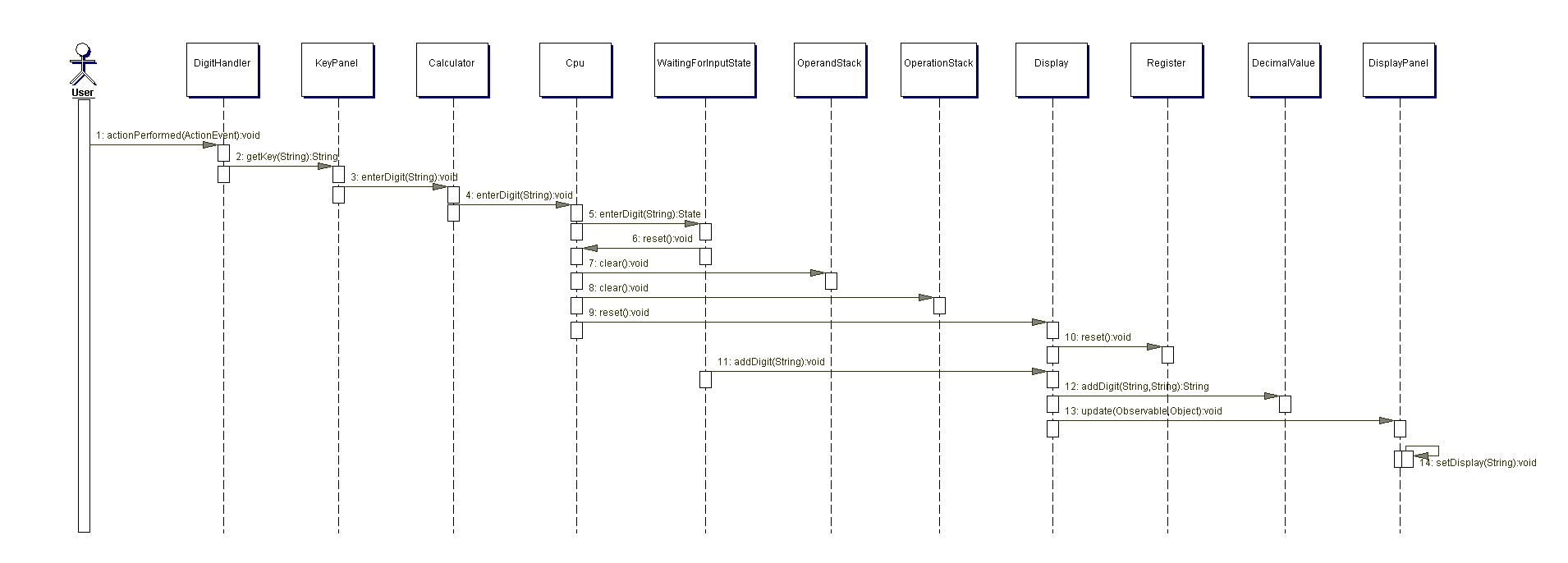
# 3.1 Use case Diagram



# 3.2 Class Diagram



# 3.3 Sequence Diagram



**4. Test Plan**



5. GitHub Repository

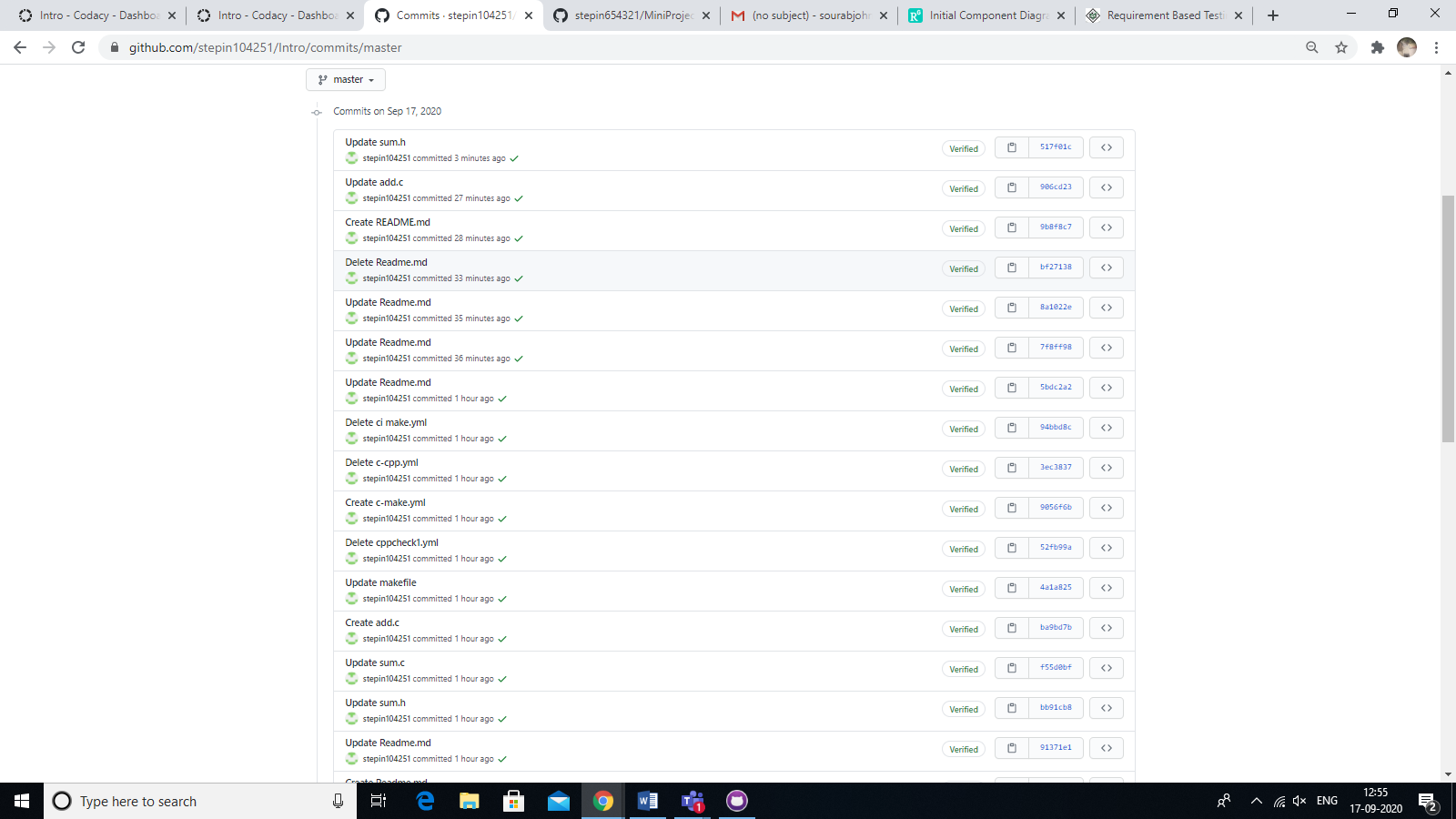
<https://github.com/99002447/Final-Submission>

6. References

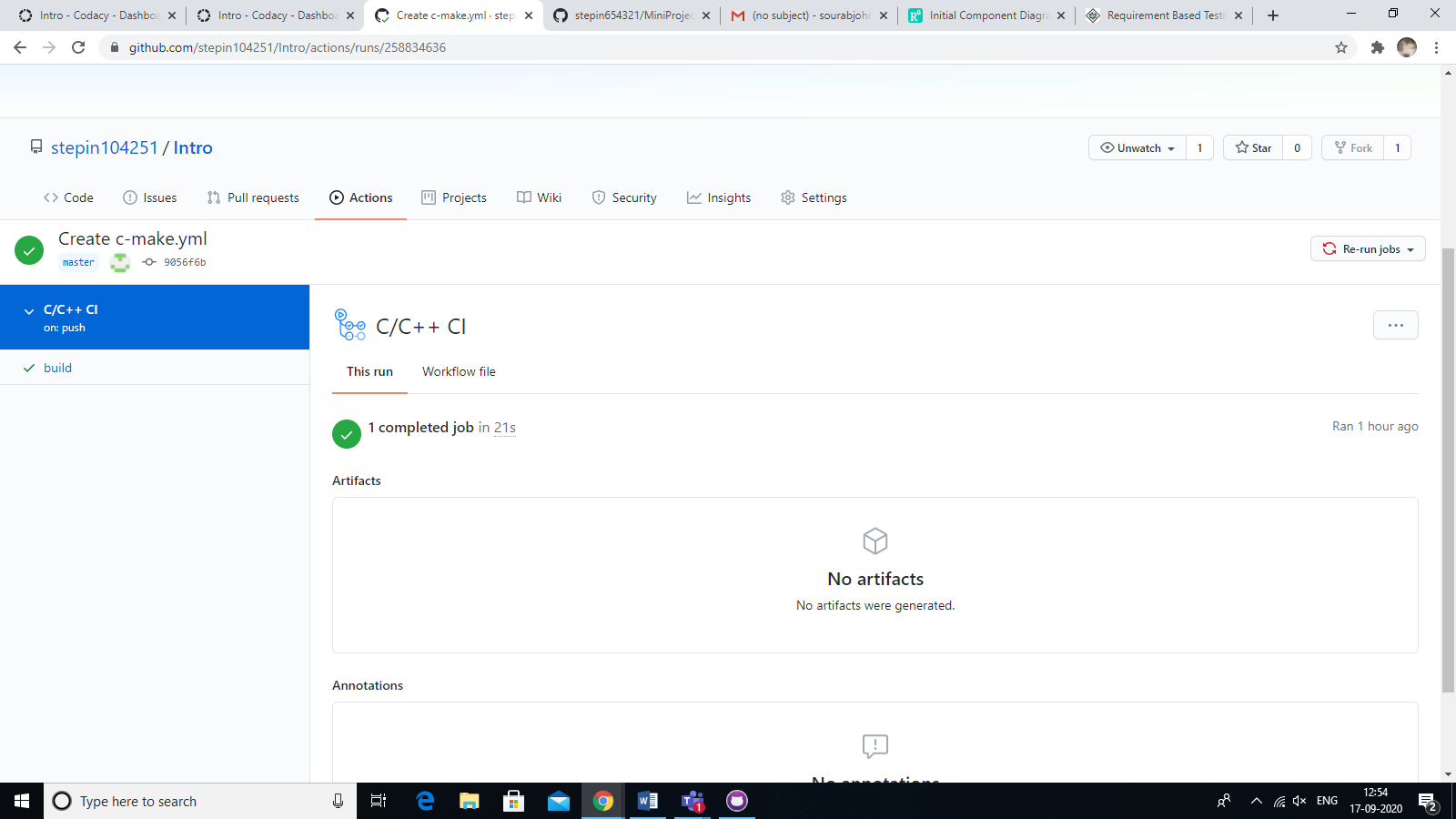
* <https://github.com/stepin654321/MiniProject_Template/tree/master/MiniProject_C>
* <https://www.includehelp.com/c-programming-questions/what-is-makefile.aspx>
* <https://www.techonthenet.com/c_language/standard_library_functions/math_h/log.php>
* <https://stackoverflow.com/questions/31861651/how-to-rename-a-directory-folder-on-github-website>

Appendix –CI Workflow for C Programming (StepIn)

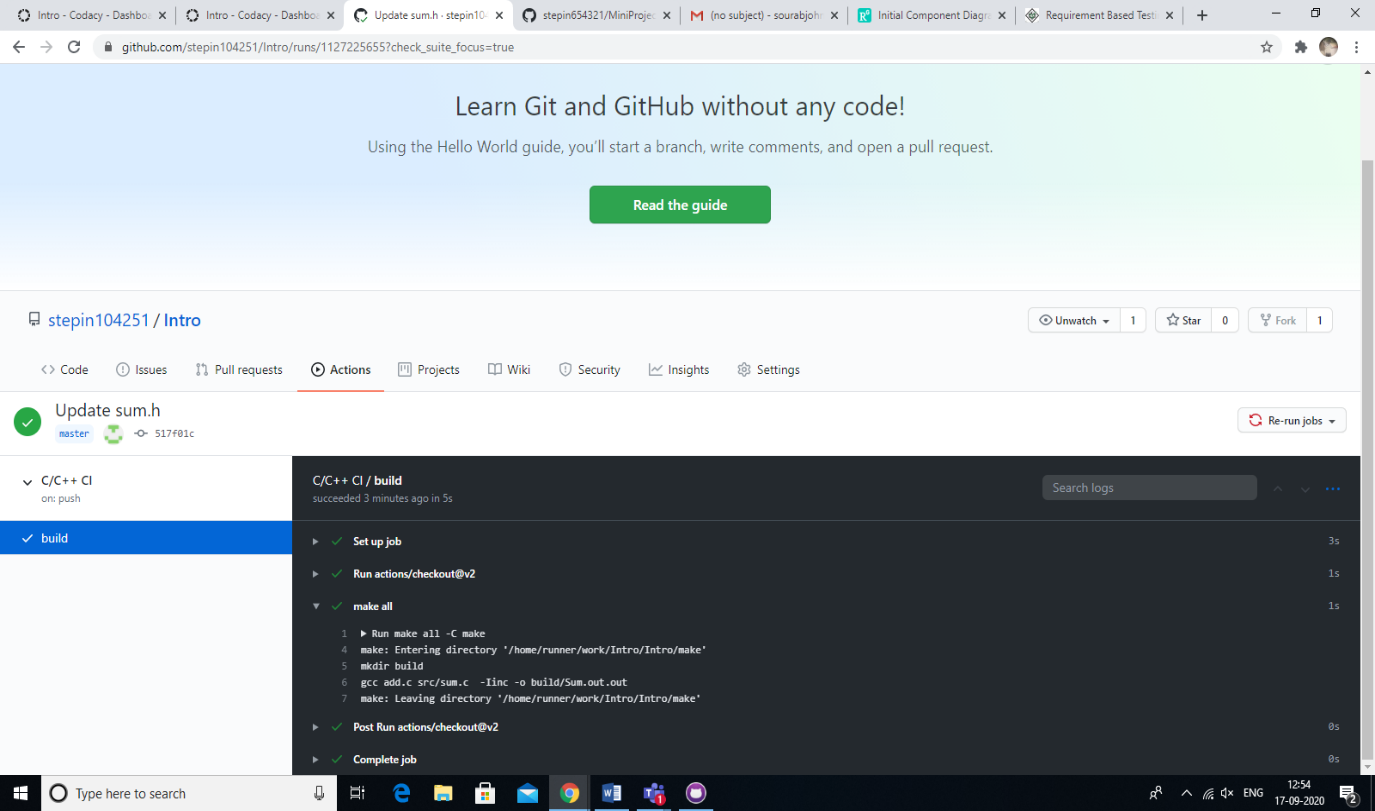
1.Git



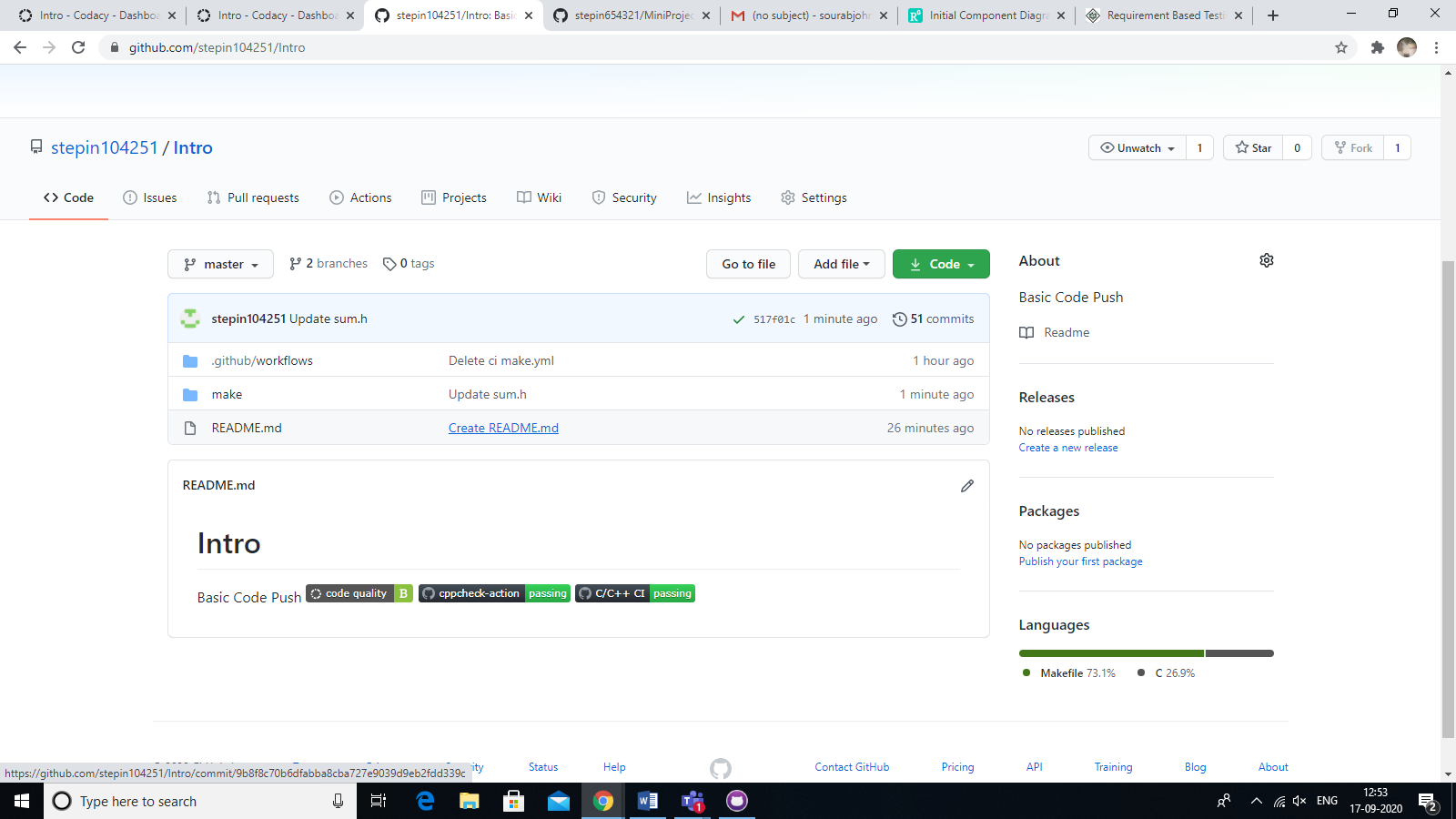
2.Make

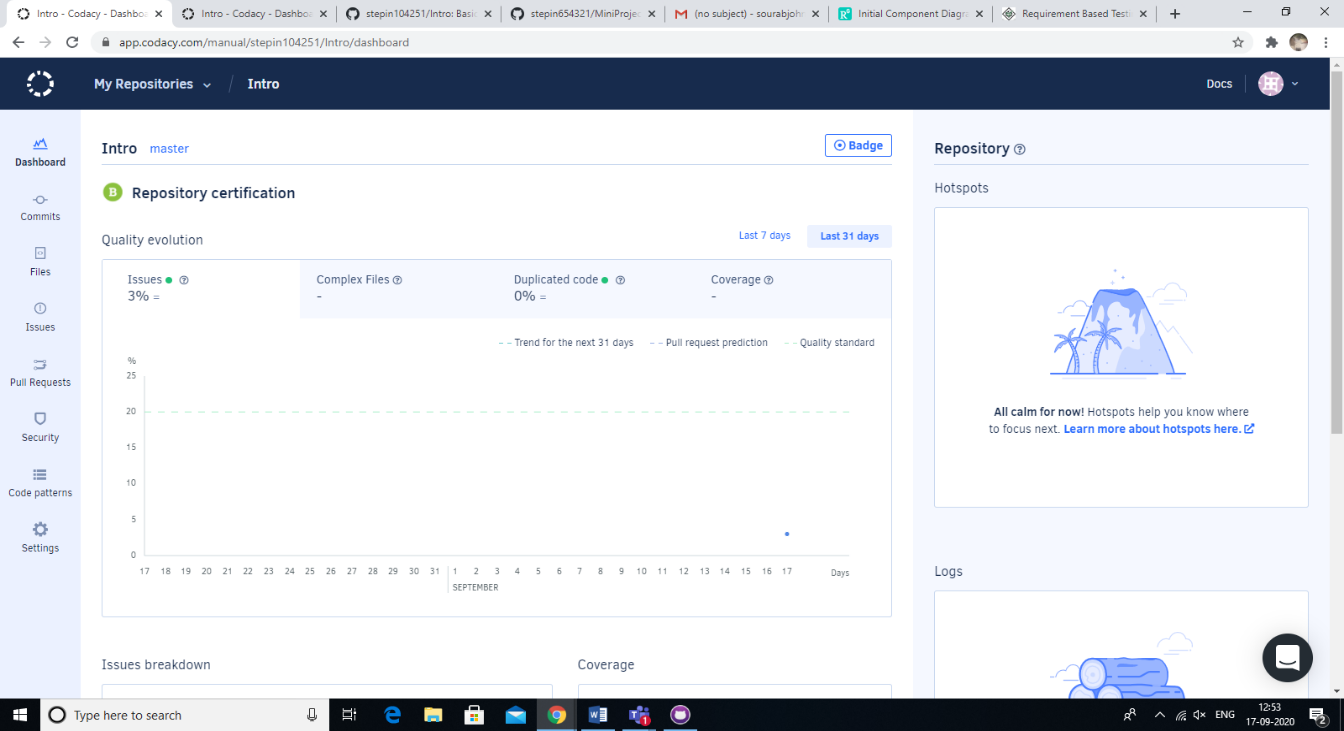


3.Build



4.Code Quality





**Link:** <http://www.github.com/stepin104251/intro>