./

Learning Report –

Day-3-

Advanced SDLC & Testing

Course Code: <CODE>



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **Approved By** | **Remarks/Revision Details** |
| 1 | 17-09-2020 | Rahul Kumar |  |  |  |
| 2 | 18-09-2020 | Rahul Kumar |  |  |  |
| 3 | 19-09-2020 | Rahul Kumar |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

./

Learning Report – Smart Trolley-Day 2

Course Code: <CODE>



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **Approved By** | **Remarks/Revision Details** |
| 1 | 17/09/2020 | Rahul Kumar |  |  |  |
| 2 | 18/09-2020 | Rahul Kumar |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

Contents For Day- 1&2

[Activity and Tasks 4](#_Toc51272040)

[**Activity 1**– System/Software Development 4](#_Toc51272041)

-Introduction

-Future scope

-Requirement Analysis

-Design

-Low & High level UML Diagram with Use-case Diagram

-Behavioural and Functional Diagrams

-High and Low level requirement

-SWAT Analysis

-Appendix

-References

[**Activity 2** –CI Workflow for C Programming 4](#_Toc51272042)

[**Activity 3** – Agile Aspects 4](#_Toc51272043)

# Checklist

* Installation of SW on Phone and Desktop
* Additional Aspects …

# Activity and Tasks

## **Activity 1**– System/Software Development

* Sub Tasks
* Complete and Evolve

## **Activity 2** –CI Workflow for C Programming

* Sub Tasks
* Complete and Evolve

## **Activity 3** – Agile Aspects

* …..

***Activity 1.1***

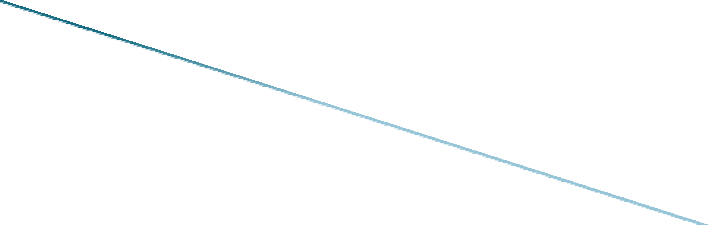
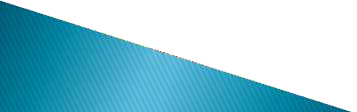
***System/Software Development***

**Introduction(Includes Cost and upgradation)**

**SMART TROLLEY-HUMAN GUIDED INTELLIGENT CART SYSTEM**

* Now a day’s shopping at big malls is becoming a daily activity in metro cities. The huge rush at malls on holidays and weekends.
* After purchase, at the billing counter the cashier prepare the bill using bar code reader which is a time consuming process and results in long queues.

The aim is to develop a system that can be used in shopping malls to solve the rush at billing counter using RFID. **BY :Rahul Kumar**



# FUTURE SCOPE :

The proposed system does not make use of intricate routing system architecture. Rather it uses simple algorithms in order to banish existing problems. Model can be further extended, to prevent the loosing of the intelligent/smart shopping cart. It can be concluded that the initial cost of the model may be high but the in subsequent years the model will be beneficial as compared to the system using barcode or manual system. Further, a more advanced micro controller, larger display module and a service to pay the bill within the cart by using swapping card can be used, thus providing the customers better services, improved consumer experience and improving time complexity to a great extent.

**Requirement Analysis:**

|  |  |
| --- | --- |
| EXISTING SYSTEM | PROPOSED SYSTEM |
| 1. Manual billing | 1.Automatic billing |
| 2. Use Barcode for billing | 2.Use RFID TAG for billing |
| 3. Human staff is needed for billing | 3.No need of any staff for billing |
| 4. Low product cost but over all expenses are much high. | 4.Product is little expensive but over-all expense is much low |
| 5. Difficult to track the product | 5.Easy to locate or track the product |
| 6. Getting product information is difficult & time consuming | 6. Getting product information is easy and no extra time needed |
| 7. It does not disclose any automatic way of indicating to shopper how the total bill is affected as the objects are added or removed from | 7. LCD or Any other Display is present which will show the updated bill every time the shopper add or remove any object from the cart. |

## 

## FUNCTIONAL REQUIREMENT:

###  Interface Requirement -The system is capable to accept and transmit the raw data which may be in the form of digital that is numeric values.

* **Audit Trail**-For each activity, the data will be recorded in the application audit trail.
* **Capacity-**The system is enough capable to hold the data and process on it.

## NON FUNCTIONAL REQUIREMENT:

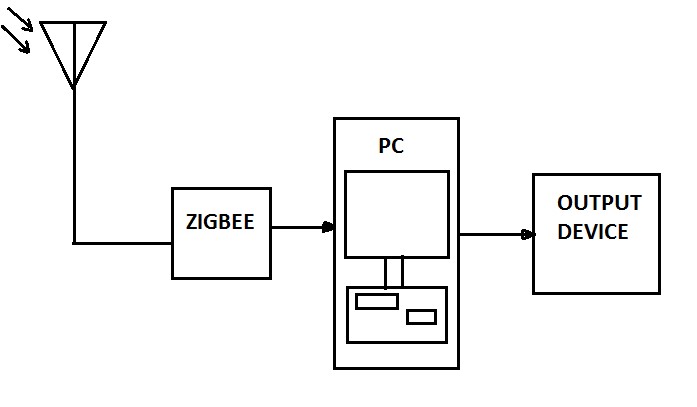
* **Maintainability:** Human resources is not required to maintain the components and collect the raw data from each of the components.
* **Reusability:** The components are compatible for changing environment and supports upgradeability.
* **Availability:** The system is functional throughout and data transfer takes place only when user requests.
* **Usability:** The system is user friendly as it uses a simple networking model like a ZigBee.
* **Reliability:** The system is highly consistent and reliable.

**Design: SYSTEM DESIGN AND IMPLEMENTATION**



**Buzzer**

***Trolley section.***

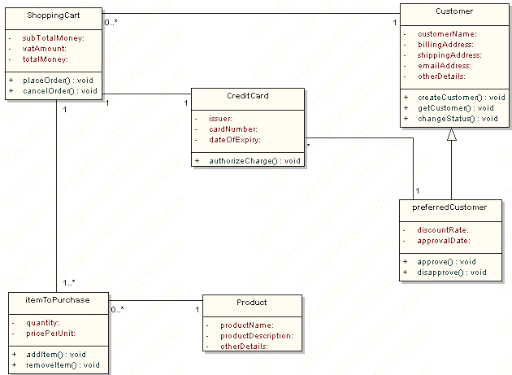


***Billing section.***

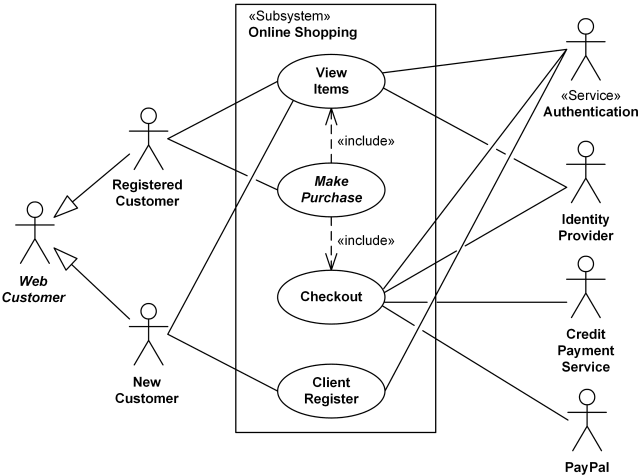
UML Diagrams:

Low Level :

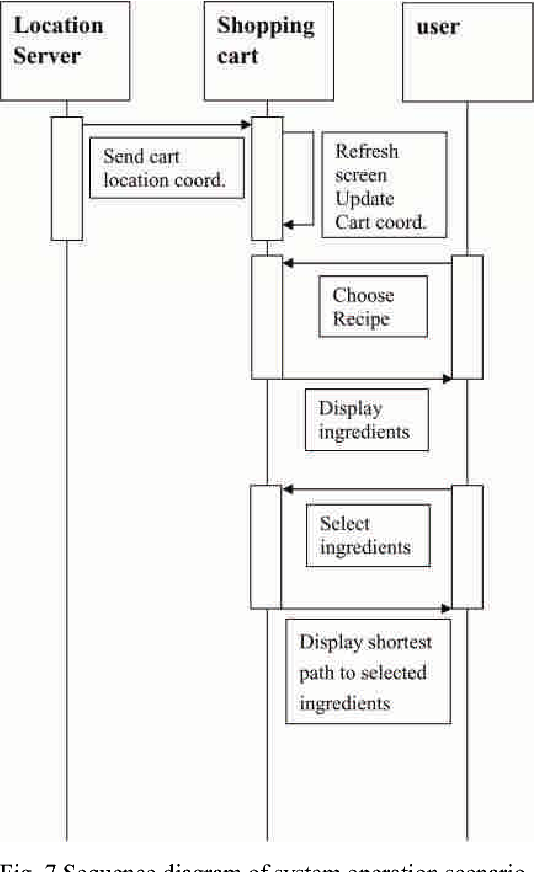
**Class Diagram:**



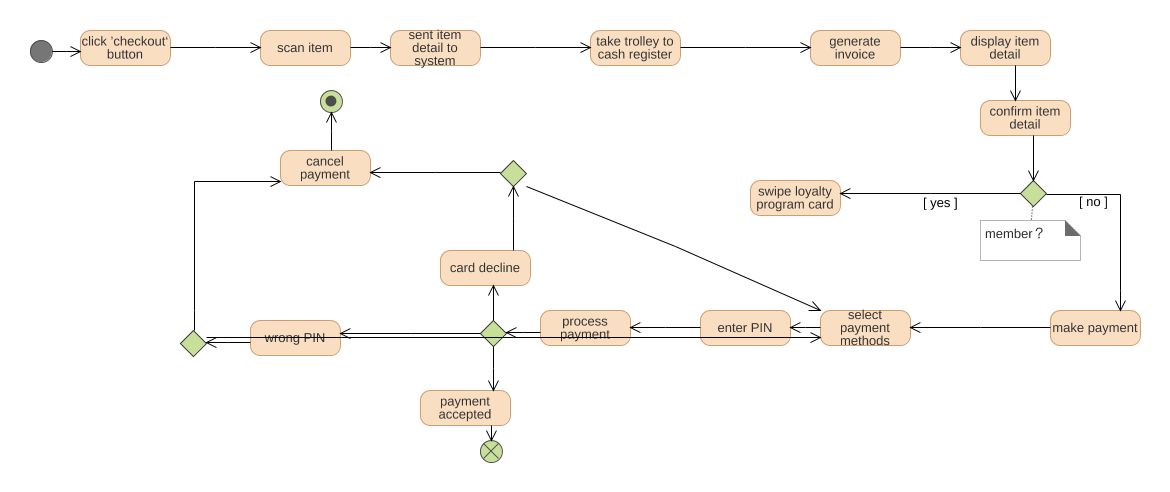
**Use case Diagram**

****

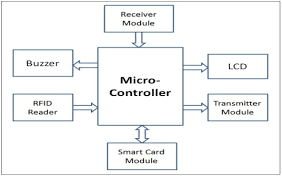
**High level :**

******

**Behavioral Diagram:**

******

**Functional Diagram:**

******

## **HIGH LEVEL REQUIREMENTS**:

|  |  |  |
| --- | --- | --- |
| **High Level Requirements ID** | **Requirements** | **Description** |
| HH\_00 | RFID Reader Module | RFID scanner module uses a RFID reader |
| HH\_01 | LCD Display | LCD (Liquid Crystal Display) screen is an electronic display module and notice a wide vary of applications |
| HH\_02 | RFID Card | A radio frequency identification reader (RFID reader) is a device accustomed gather data from an RFID tag that is employed to trace individual objects |
| HH\_03 | Arduino uno | Has twenty digital input/output pins for different operations |
| HH\_04 | Microcontroller | Used for assembling all the other part for controlling purpose. |
| HH\_05 | Comfortable Design | The Trolley should be comfortable to handle for longer duration. |
| HH\_06 | Long battery backup | The Trolley should be able to work without charging or replacing the battery for about 8 hours. |
| HH\_07 | Good functional parts | To provide desirable and smooth functioning. |

## **LOW LEVEL REQUIREMENTS**:

|  |  |  |
| --- | --- | --- |
| **Low Level Requirements ID** | **Requirements** | **Description** |
| HH\_00\_LL\_00 | RFID Reader Module | RFID scanner module uses a RFID reader |
| HH\_00\_LL\_01 | Well desinged algorithms | To compute error-free functioning. |
| HH\_00\_LL\_02 | Mobile and Mobile app | For controlling |
| HH\_06\_LL\_03 | Low power consumption | To enhance the battery backup. |
| HH\_02\_LL\_04 | Control buttons on Trolley | User friendly feature and ergonomic design for emergency. |
| HH\_07\_LL\_05 | Buzzer | To provide good sound quality within safe limits in emergency. |
| HH\_03\_LL\_06 | Bluetooth Connectivity | For wireless connection. |
| HH\_05\_LL\_07 | Wi-fi Module | For comfortable in connecting |
| HH\_04\_LL\_08 | Designing for all | Comfortable for all . |

**High level Test Plan:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Precondition** | **Expected Input** | **Expected Output** | **Actual Output** |
| **001** | ****Cart – Backend sync validation**** | * Verify that the correct PDP page is shown with all the necessary   ) | Select any product whose inventory count is less any (to be validated at backend) | Added to Cart icon for the Product. | Try Editing the quantity of the product in cart with a 5 and try checkout |
| **002** | ****Single Product Remove from Cart**** | * Verify that the correct PDP page is shown with all the necessary | Select any product. | Added to Cart icon | Shown in the Cart icon |
| **003** | ****Multiple Products Remove from Cart**** | Select any product | Add to Cart icon for the First Product | Added to Cart icon for the Second Product | Cross symbol to remove the any of the product from the cart. |
| **004** | ****Multiple Products Remove from Cart**** | Select any product | Remove items from Cart icon for the First Product. | Removed Cart icon for the Second and third Product. | Products should be removed from the cart and Cart icon should show only 1 items. |
| **005** | ****Edit Quantity Cart boundary value**** | Select any product. | Try editing the quantity of the item in cart with a value greater than 5 and try checkout. | * Verify that product should be successfully added to cart. | * Verify that if the items quantity count is greater than 5 then checkout is not allowed and an appropriate error should be presented to the user for the same. |

|  |
| --- |
|  |

**Low level Test Plan:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Precondition** | **Expected Input** | **Expected Output** | **Actual Output** |
| **001** | **Buy Now** | * Verify that the correct PDP page is shown with all the necessary links and information. | Select any product. | Selected any product. | Buy Now option for selected |
| **002** | ****Guest Check-out**** | Launch the test app. Do not sign in. | Select any product to purchase | Product added to purchase | Verify that user should successful checkout the product |
| **003** | ****Registered Check Out**** | Launch the test app. Sign in. | Select any product to purchase. | Added the product to purchase. | * Verify that user should successful checkout the product. |
| **004** | ****Check-Out Mandatory Field Validation.**** | Launch the test app. Sign in. | Select any product to purchase | Added the product to purchase. | Try completing the check-out process by leaving any mandatory field blank in shipping or billing address. |
| **005** | **Order Confirmation Order Id Validation** | Do checkout for any of the product. | Do checkout for any of the product. | Checkout automatically | * Verify that the order id shown in the Order confirmation message should be correct and OMS should have entry for that. |

**SWOT Analysis:**

* Strengths: Cost and time effective, Real-time System.
* Weakness: Difficult Integration.
* Threats: RFID tag System, Web & Smartphone Application (Cyber Attacks).
* Opportunities: Fast lane check-out (can be more time effective), Smart Wallet or online payment using mobile app or web application.

**Appendix:**

**https://github.com/stepin105329/testsum**

**REFERENCES:**

- > CONTROL THERORY& INFORMATICS ISSN 2224- 5774(PRINT) ISSN2225-0492(ONLINE), VOL1, NO1, 2011 RFID Based Automatic Shopping Cart SAURABH KUMAR SULTAINA, GOURAV JAISWAL, PRATEEK JAIN .

->INTERNATIONALCONFERENCE ON COMPUTING AND CONTROL ENGINEERING (ICCCE2012), 12&13 APRIL,2012 .

->Smart Shopping Experience Based On RFID.VANITHA

[THE SEVENTH INTERNATIONAL CONFERENCE ON SENSOR TECHNOLOGIES &APPLICATIONS Smart Shopping Cart For Automated Billing Purpose Using Wireless Sensor Networks, SANCHITA ROY,UDITA GANGWAL,JYOTSNA BAPATRobert Bakker, Edwin Keijsers, and Hans van der Beak “Alternativ.

->Dr.Suryaprasad J, Praveen Kumar B O, Roopa D Arjun A K, A Novel Low-Cost Intelligent Shopping Cart, Proceedings of the 2nd IEEE International Conference on Networked Embedded Systems for Enterprise Applications, NESEA 2011, Perth, Australia, December 8-9, 2011 .

->Swati Zope, Prof. .Maruti Limkar, “RFID based Bill Generation and Payment through Mobile”, International Journal of Computer Science and Network (IJCSN), Volume 1, Issue 3, June 2012.

-> G. Roussos and B. College, “Enabling RFID in Retail”, Computer, IEEE, vol. 39, no. 3, 2006, pp. 25-30. [7] Cisco Internet Business Solution Group survey document, My Shopping My Survey Findings, http://www.cisco.com/web/about/ac79/docs/retail/Mashop-surveymetricsUNITED STATES.pdf [8] American Time Use Survey <http://www.bls.gov/tus/charts/>.

->http:// [www.rfidjournal.com](http://www.rfidjournal.com)

-> http://en.wikipedia.org/wiki/zigbee .

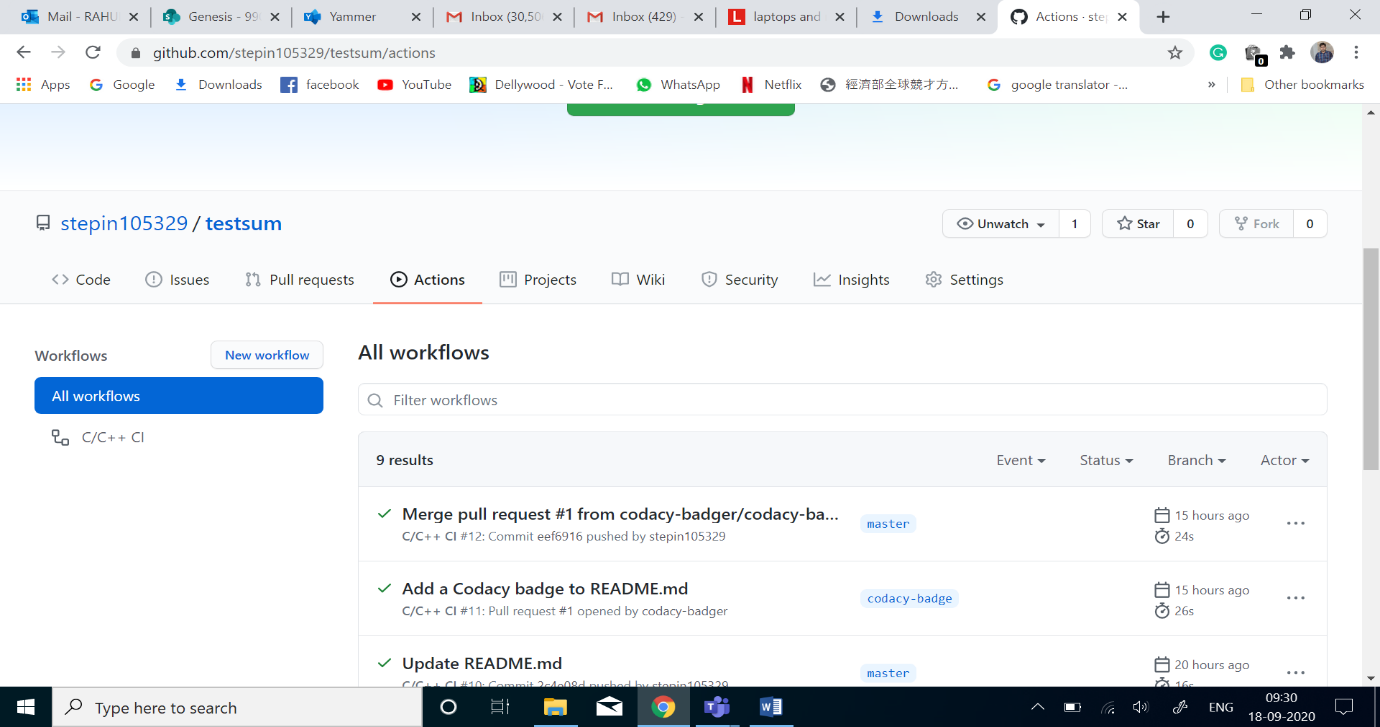
->Mazidi, Mazidi, McKinlay- “The 8051 microcontroller and Embedded Systems”.

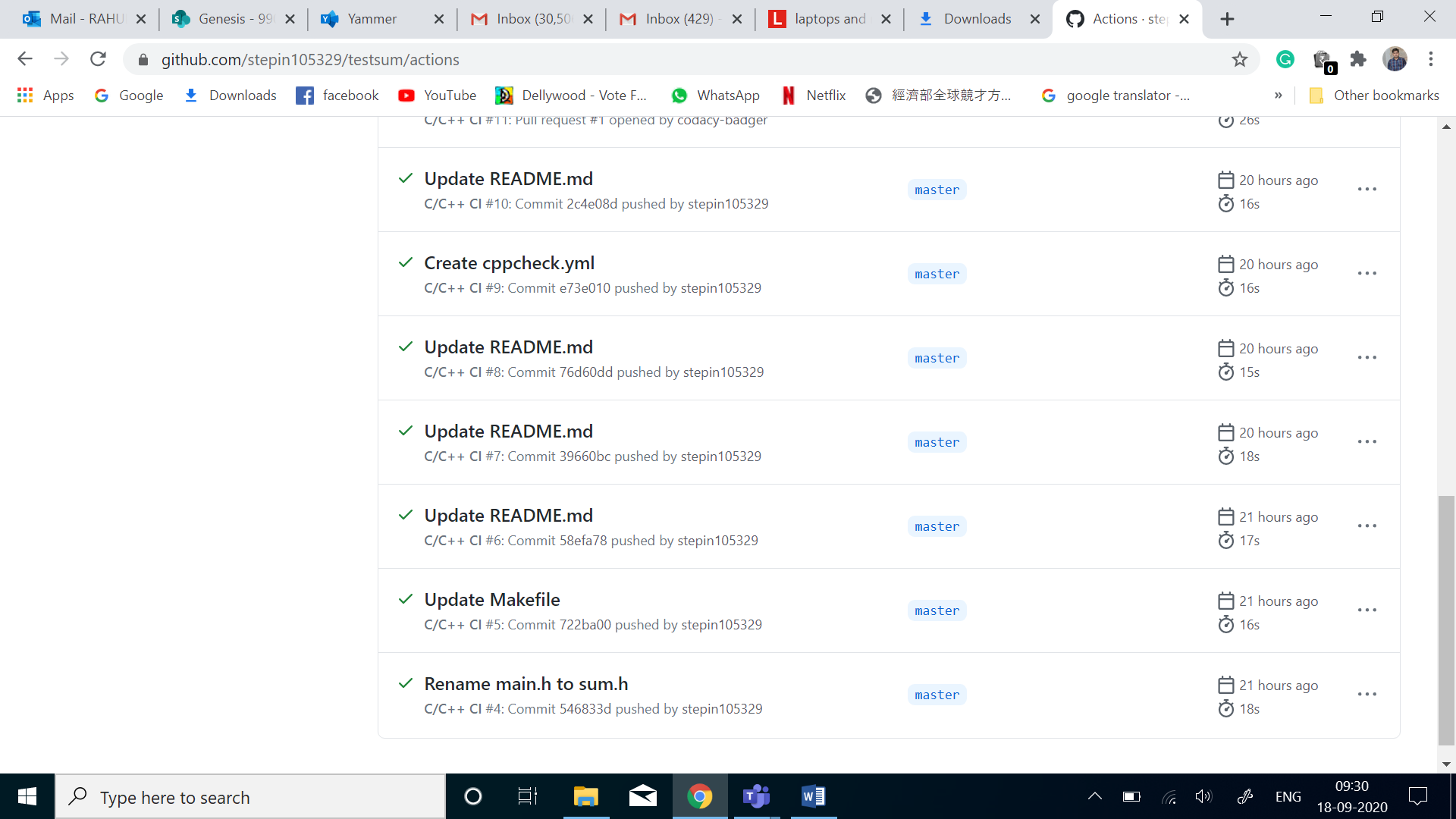
-> Savi Technologies. ”Active and Passive RFID: Two Distinct, But Complementary, Technologies for Real-Time Supply Chain Visibility, SAVI Technology

***Activity -1.2***

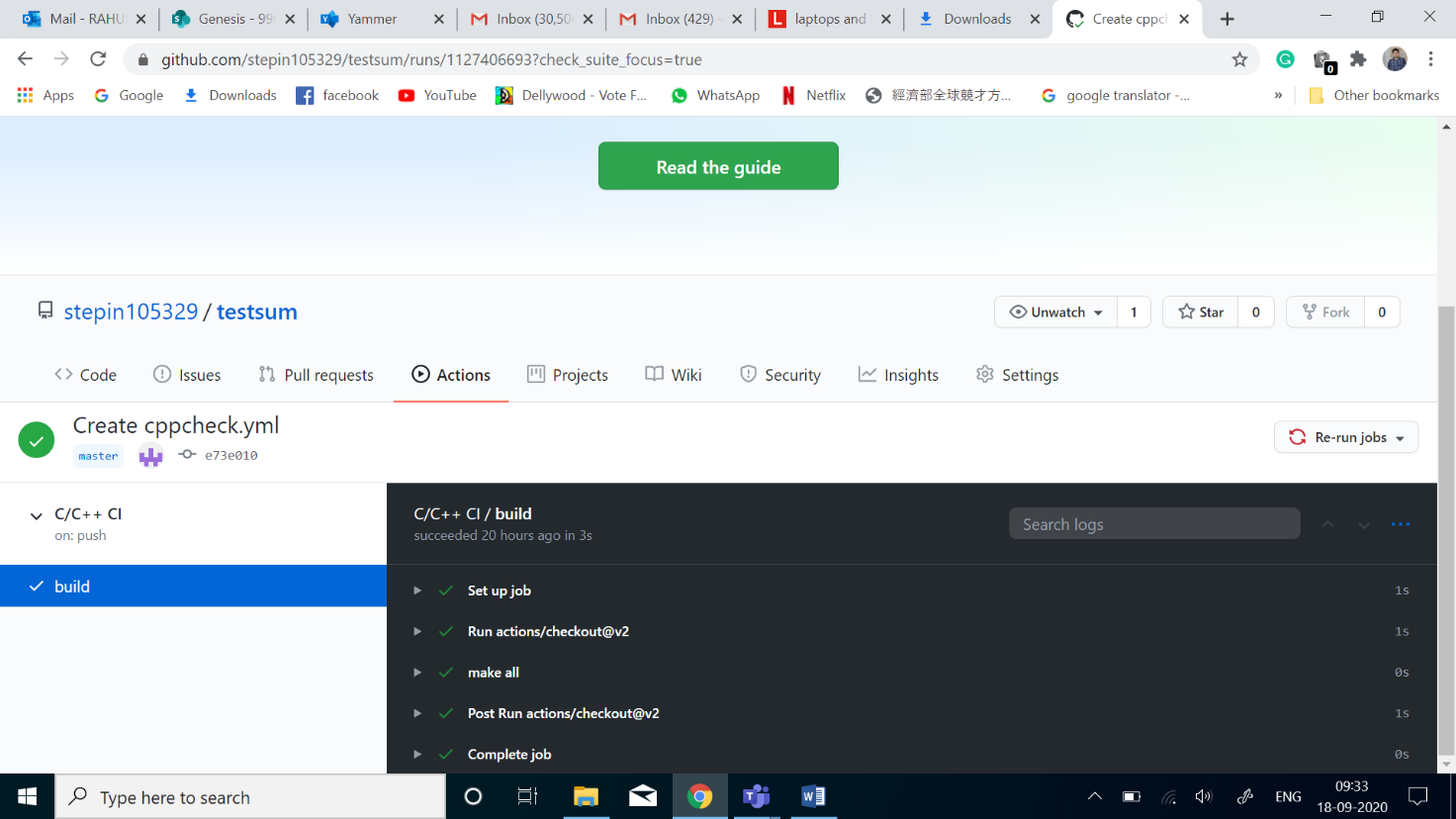
*CI Workflow for C Programming*

[**Github repo**](https://github.com/stepin105343/stepin-practice)

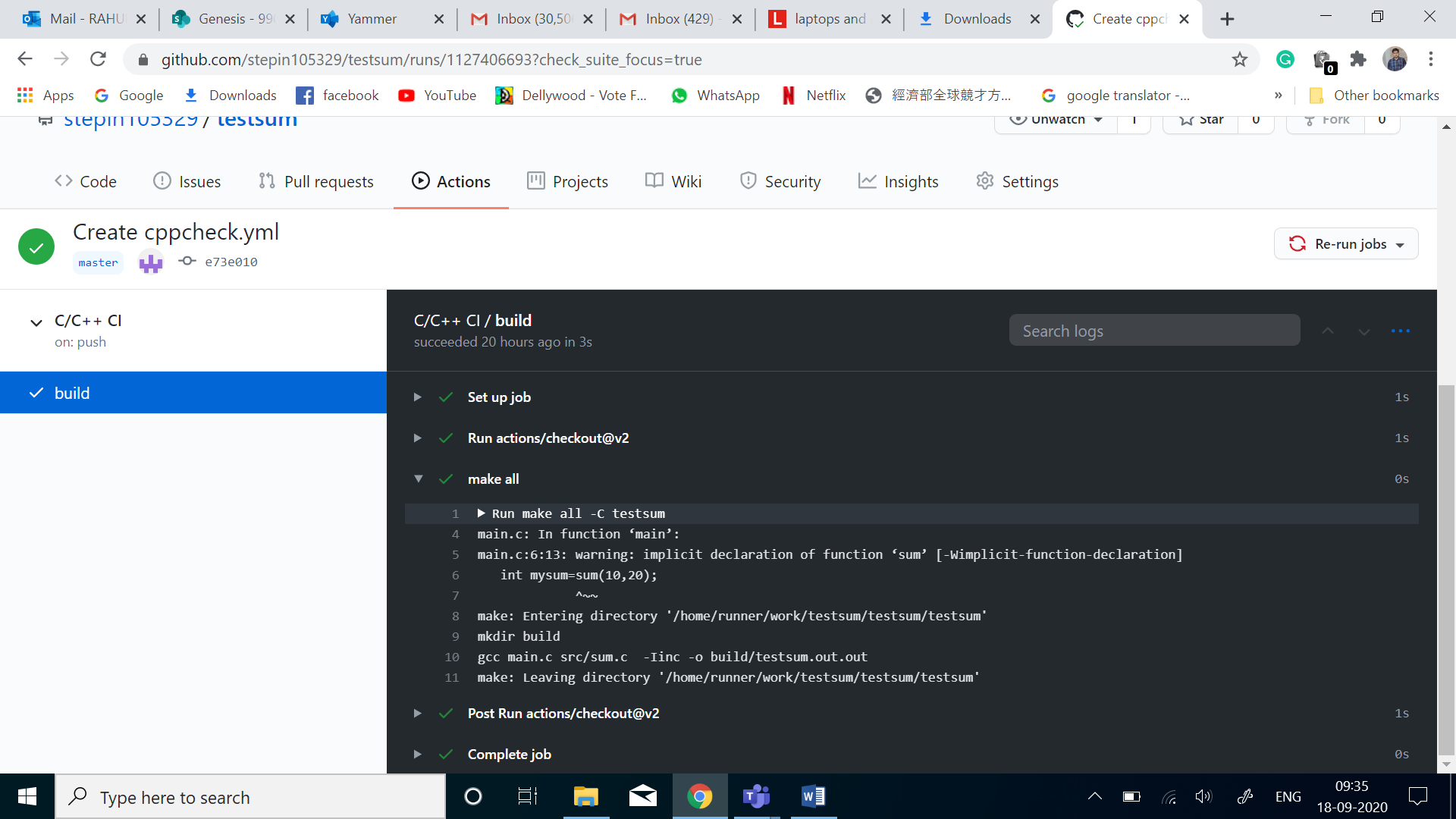
**Actions:**



**Build:**



**Makefile:**

****

|  |
| --- |
| SRC = unity/unity.c\ |
|  |

|  |
| --- |
| src/CheckPrime.c\ |
|  |

|  |
| --- |
| test/Test\_CheckPrime.c\ |
|  |

|  |
| --- |
| main.c |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| INC = -Iunity\ |
|  |

|  |
| --- |
| -Iinc\ |
|  |

|  |
| --- |
| -Itest |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| PROJECT\_NAME = CHECK\_PRIME.out |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| $(PROJECT\_NAME): $(SRC) |
|  |

|  |
| --- |
| gcc $(SRC) $(INC) -o $(PROJECT\_NAME) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| run:$(PROJECT\_NAME) |
|  |

|  |
| --- |
| ./${PROJECT\_NAME} |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| doc: |
|  |

|  |
| --- |
| make -C documentation |
|  |

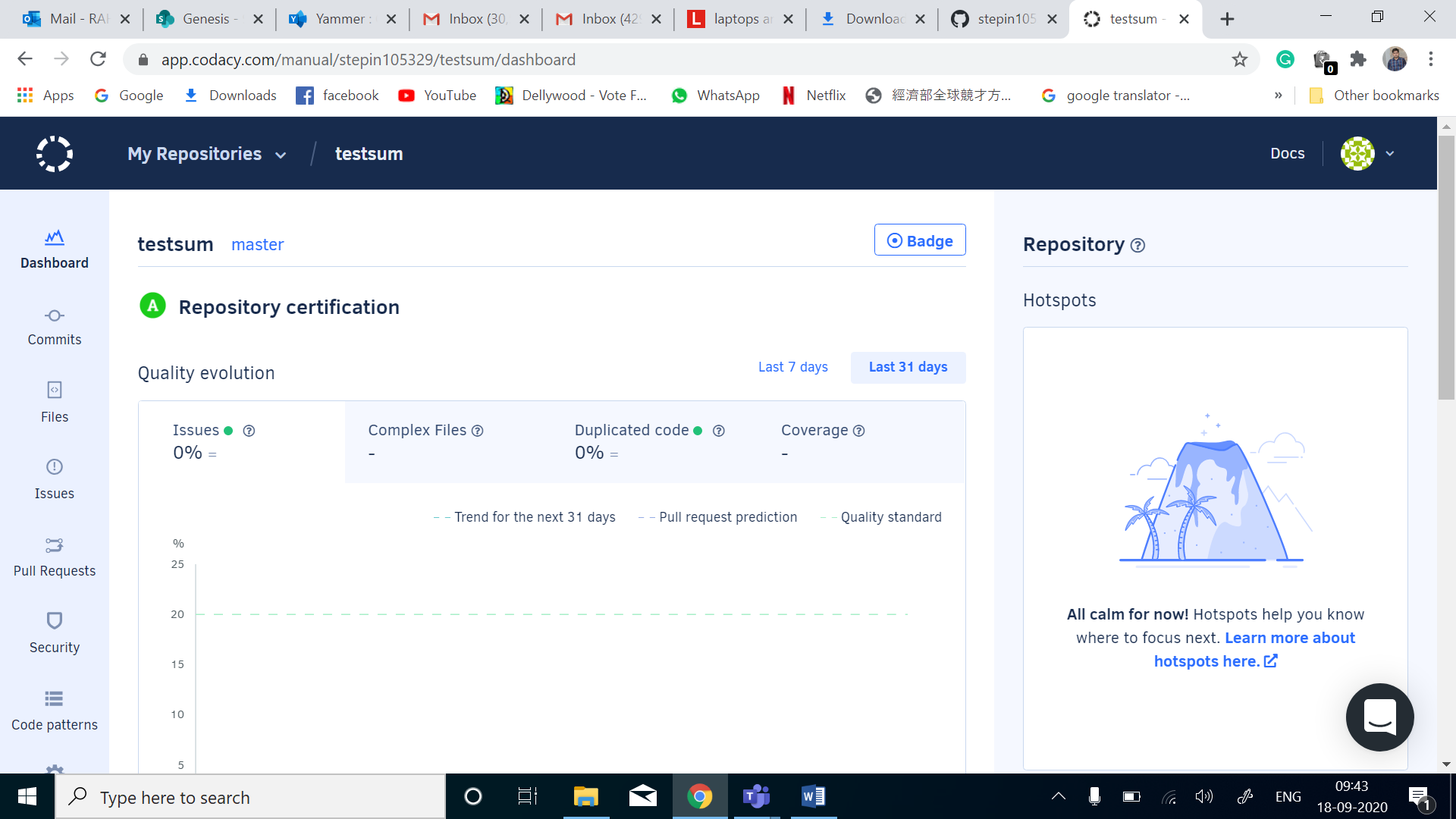
|  |
| --- |
|  |
|  |

|  |
| --- |
| clean: |
|  |

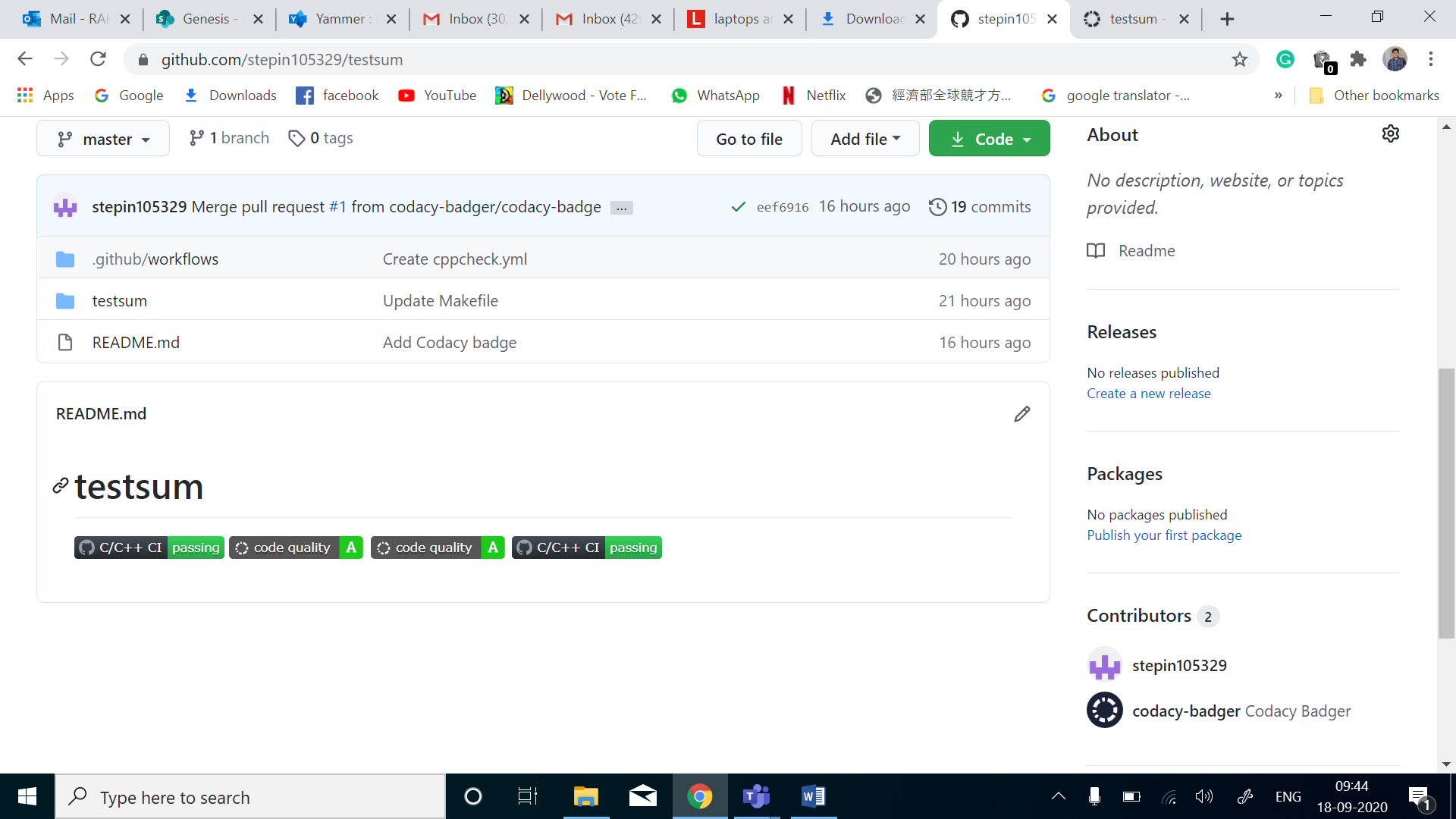
rm -rf $(PROJECT\_NAME) documentation/html

**Code quality**

**Codacy details:**



**Badges for all the above Acticities:**

****

**APPENDIX:**

[**https://github.com/stepin105329/testsum**](https://github.com/stepin105329/testsum)

***Activity-2***

**Agile Aspects:**

**THEME:**

To design a smart trolley which indicates the amount of every information required for shopping with the help of smooth functioning and indicates the user after completion of shopping to the electronic device.

**EPIC:**

1) The sensors working will give the appropriate output to the user:

If the sensors are in working condition then the suggestions for the quantity and indicators will be assigned and also reduces the malfunctioning of the product.

2) The display system and microcontroller working:

Proper display and proper message is transferred when the above components are working good.

**USER STORIES:**

The Smart Trolley should enable display of process of GUI to the electronic device (mobile).

The Smart Trolley should placement of goods for proper suggestions of conditions.

The Smart Trolley should be of utmost safety during use like overloading.

The Smart Trolley should give proper outputs after sensing the external environment.

The Smart Trolley should give the message to electronic device without any error.

The Smart Trolley should display the area of concern during mal-function.

**Comparisons:**

|  |  |  |
| --- | --- | --- |
| Parameters | Agile Model | Non-Agile Models |
| Approach of this methodology | This methodology is very flexible and adjustable and can adapt to the project needs. | This methodology is not as flexible as Agile model and it’s tough to accommodate changes in the project. |
| Measurement of Success | The success of the project in Agile model is measured by the Business value delivered. | In this methodology the success of the project is measured by the Conformation to plan. |
| Size of the Project | The Project size is usually small in Agile model. However larger projects can also be handled using the **Scaled Agile Framework (SAFe)**. | The project size is Large in non- Agile models. |
| Style of Management | The style of management in Agile model is not centralized. It is distributed among the team members. | The management style in the non-Agile models is dictatorial. Only one person is the decision maker and rest of the people follows him. |
| Ability to adapt to change | In Agile model the changes are accepted and adapted as per the project needs. | But in non-Agile models the changes are not accepted easily in the later stages of the development. |
| Documentation required | Less documentation is required in Agile. | More documentation is required in non-Agile models. |
| Importance of | In Agile model more emphasis is given to the people that means it’s People- Oriented. | In non-Agile models the more importance is given to the process hence it’s Process- Oreinted. |
| Cycles or iterations | Agile methodology has many cycles or iterations which is also known as Sprints. | But, in Non-Agile methodology the cycles are limited. |
| Planning in Advance | There is minimal upfront planning in Agile methodology. | In Non-Agile models the planning should be complete before the development starts. |
| Revenue | In Agile method the return on investment is early in the project cycle. | In non-Agile methods the return on investment is at the end of the project. |
| Size of the team | The size of the team in Agile methodology is usually small and creative. | But in Non-Agile models the team size is large. |

## **When to use Agile and Non-Agile models:**

|  |  |  |
| --- | --- | --- |
| **Project Attributes** | **Agile Model** | **Non-Agile Model** |
| Requirement of the Project | Requirements in Agile model can change as per the customer requirement. Sometimes requirements are not very clear. | In Non-Agile models the requirements are very clear before entering into the development phases. Any change in the requirement is not easily accepted during the development phases. |
| Size of the Project | The Project size is small in Agile model hence small team is required. | But in Non-Agile models the Project size is usually big hence big team is required. |
| Design of the Project | In Agile model the architecture is made as per the current requirements but is designed to be flexible. | In Non-Agile models the architecture is made as per the current requirements as well as for future requirements. |
| Planning and Control of the Project | In Agile model the planning of the project is Internalized and has qualitative control. | But in Non-Agile models the plans are documented properly and have quantitative control. |
| Type of Customers | Agile methodology is followed by the collaborated, dedicated collated and knowledgeable customers. | In Non-Agile models the customers are of Contract provisions. |
| Developers required | In Agile model the developers should be knowledgeable, analytically strong, collated and collaborative. | In Non-Agile models the developers should be more Plan Oriented. |
| Refactoring | In Agile model refactoring is not costly. | But in Non-Agile models the refactoring is very costly. |
| Risks involved | Usually in Agile models the chances of occurrence of unknown risks are more which can have major impact in the project. | In Non-Agile models the risks are understood clearly and the impact of the risk in the project is very less. |

# 

# 

# Activity-3,DAY-3

Contents For Day-3

[Checklist 3](#_Toc51272039)

[Activity and Tasks 4](#_Toc51272040)

[**Activity 1**– System/Software Development 4](#_Toc51272041)

->Introduction

->Design

-> Requirement Analysis

-> UML Diagrams (Design)

-> Test Plan

[**Activity 2** –CI Workflow for C Programming 4](#_Toc51272042)

->Commits

->Make

->Cpp Check

->Build

->Unit-testing

->Code Quality (Codacy)

->Appendix

->References

# Checklist

* Installation of SW on Phone and Desktop
* Additional Aspects …

# Activity and Tasks

## **Activity 1**– System/Software Development

* Sub Tasks
* Complete and Evolve

## **Activity 2** –CI Workflow for C Programming

* Sub Tasks
* Complete and Evolve.

### INTRODUCTION:

### LMS is a small program to manage library.

### Adding new users, books and magazines/newspaper.

### Updating users, books and magazines/newspaper.

### Removing users, books and magazines/newspaper.

### Issuing books and collecting books back.

### Searching for books.

### Keep track of all information.

### Easy to use.

### Information will be stored in database.

**Requirement Analysis:**

### Functional requirements:

### -A new user which is not registered in the system.

### - Registration form must be available.

### -Details must be correct.

### -Member must be registered against unique ID

### - Books must be available

### -We must have librarian account which manages the whole system.

**Non-Functional Requirements:**

### User authentication and validation of members using their unique member ID.

### Proper accountability which include not allowing a member to see other members account.

### Only administrator will see and manage all members account.

### CAPTCHA words will be used for user login.

### Proper user authentication should be provided.

**HIGH LEVEL REQUIREMENTS**

|  |  |  |
| --- | --- | --- |
| **ID** | **REQUIREMENT** | **DESCRIPTION** |
| HH\_01 | OPERATING SYSTEM | It is used for Proper  Operation of library management |
| HH\_02 | DATABASE | Used for retrieving of records |
| HH\_03 | RAM | For storing the records |
| HH\_04 | PROCESSOR | For developing and maintaining project |

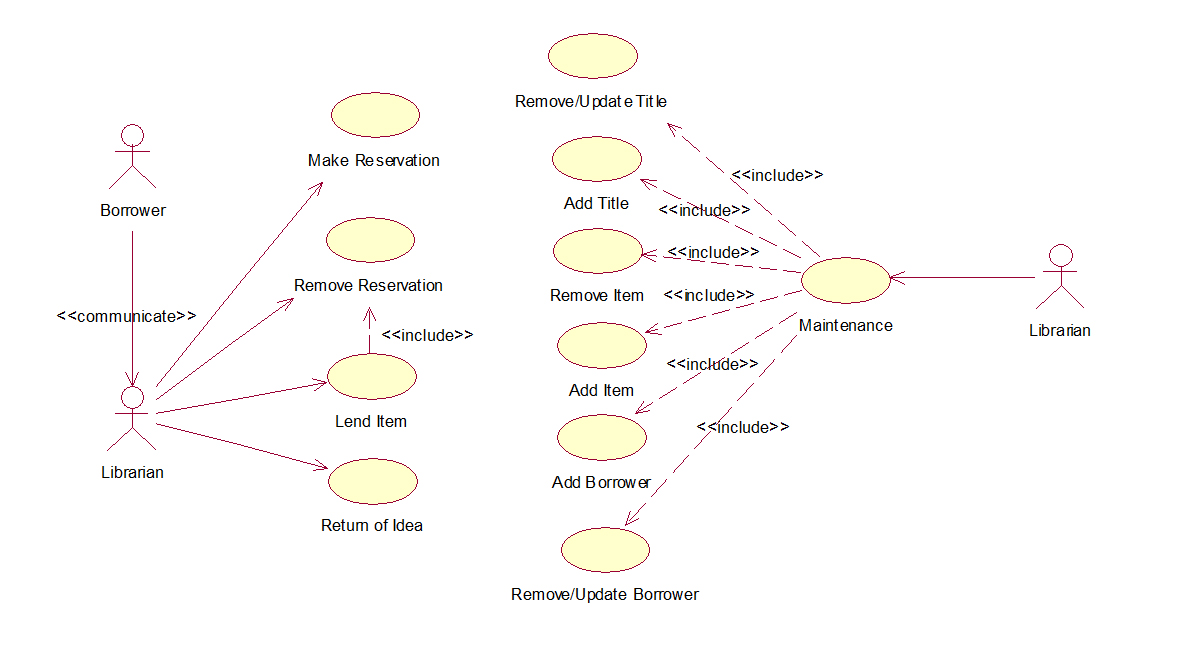
**LOW LEVEL REQUIREMENTS:**

|  |  |  |
| --- | --- | --- |
| **ID** | **REQUIREMENTS** | **DESCRIPTION** |
| **HL\_01** | **LIBRARIAN** | For checking ,updating delete books and seek permission |
| **HL\_02** | **MEMBER** | Search and request book and also search account |
| **HL\_03** | **GUEST** | For new enrollments in the database |

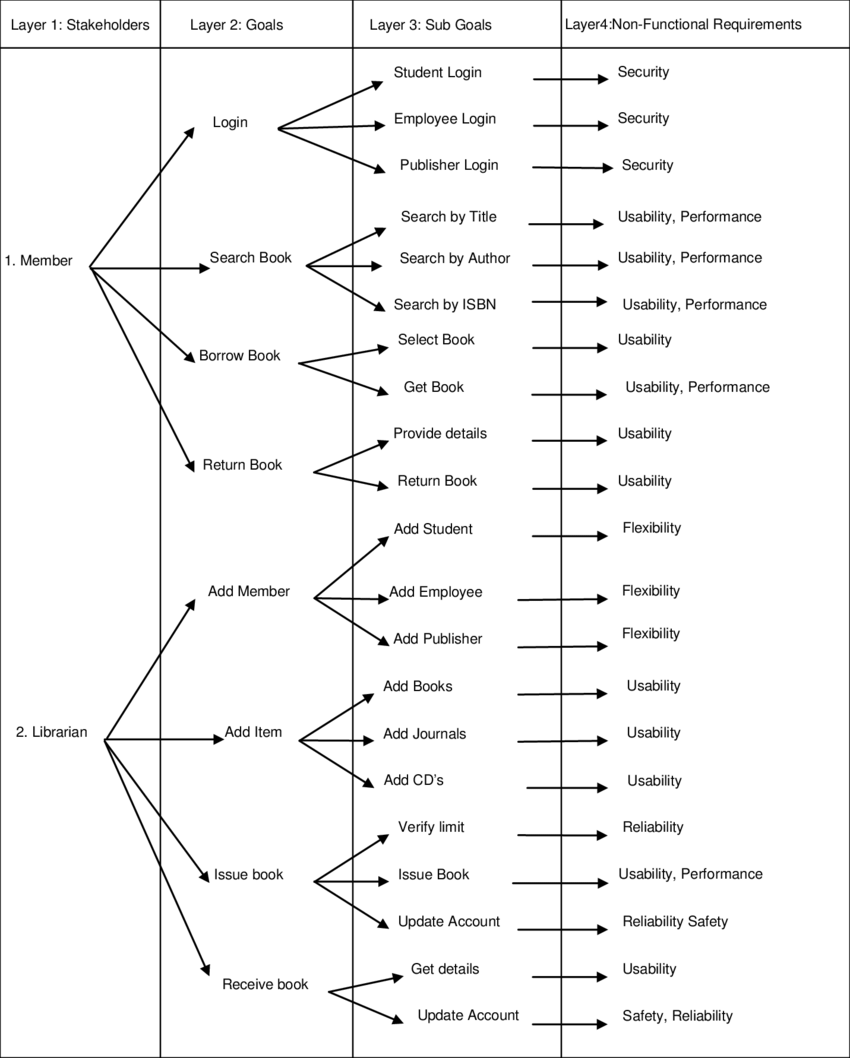
**Design**

**UML Diagrams:**

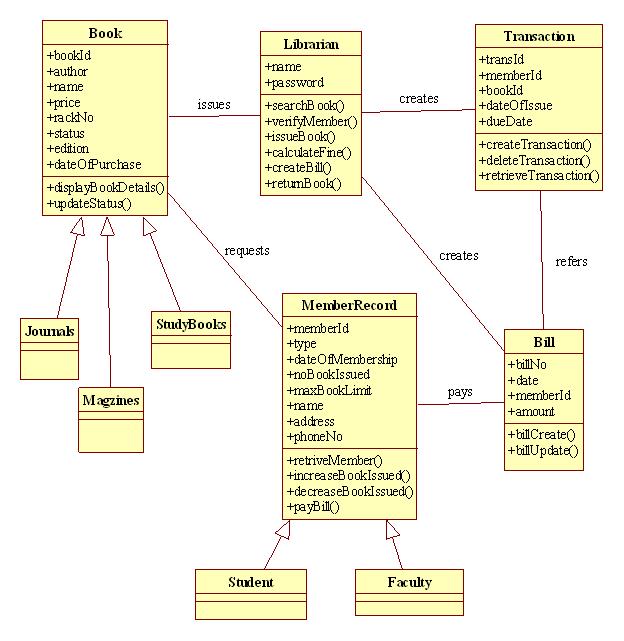
**Use-Case Diagram:**

****

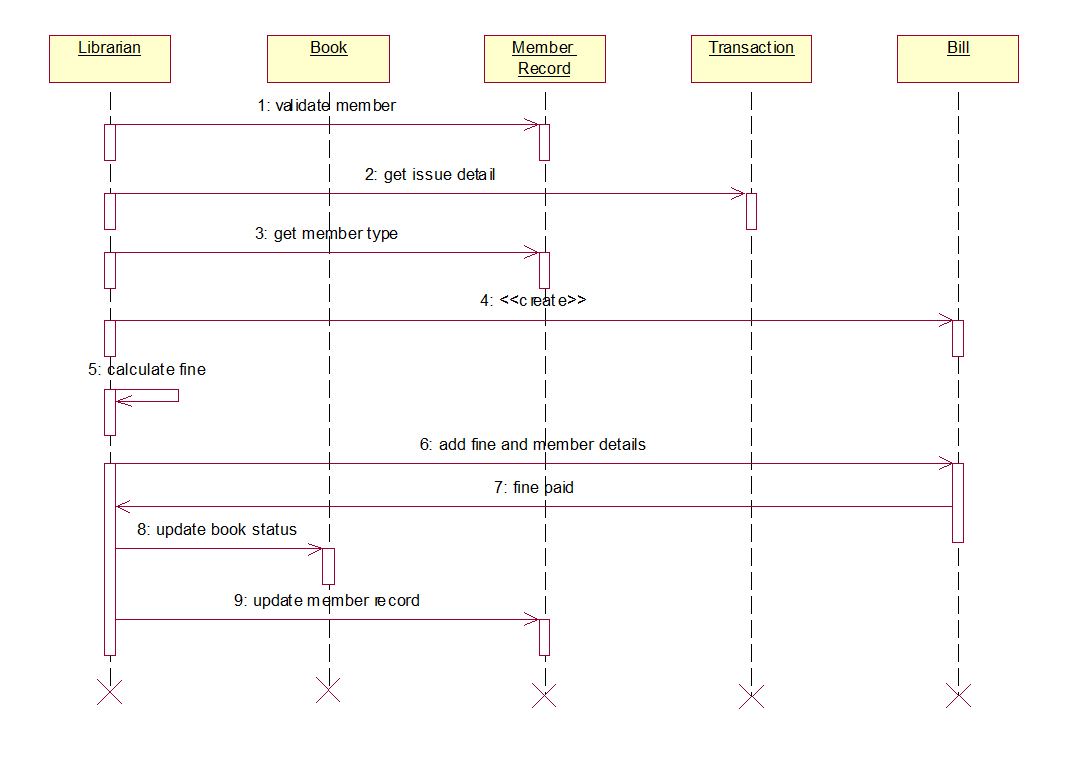
**Design:**



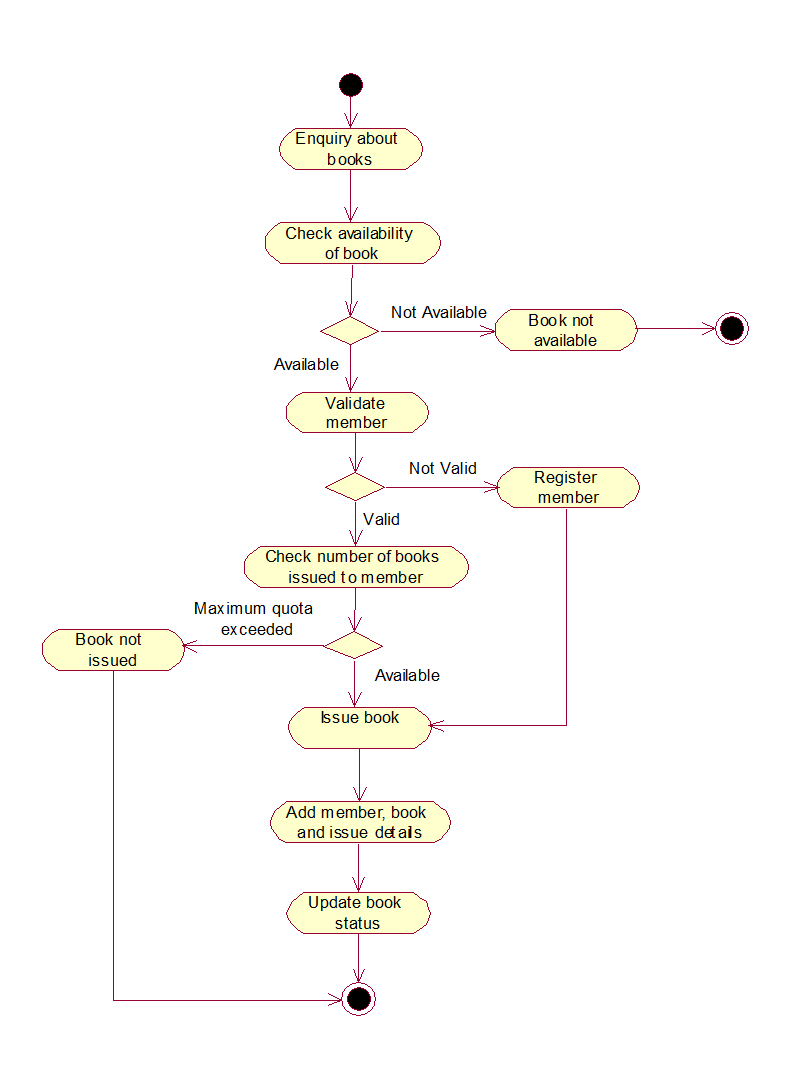
**Class Diagram:**



**Sequence Diagram:**



**Activity Diagram:**

****

**Test Plan:**

**High level Test Plan:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Precondition** | **Expected Input** | **Expected Output** | **Actual Output** |
| **001** | On the click of ADD button | At first user have to fill all fields with proper data , if any Error like entering text data instead of number or entering number instead of text.is found then it gives proper message otherwise Adds Record To the Database | Click on add Button | successful | successful |
| **002** | On the Click of DELETE Button | This deletes the details of book by using Accession no. | Click on Delete Button | successful | successful |
| **003** | On the Click of UPDATE  Button | Modified records are Updated in database by clicking UPDATE button. | Click on Update Button | successful | successful. |
| **004** | On the Click of SEARCH Button | Displays the Details of book for entered Accession no. Otherwise gives proper Error message. | Click of SEARCH Button | successful | Successful. |
| **005** | On the Click of EXIT button | Exit the current book details form | Click of EXIT button | successful | successful |

|  |
| --- |
|  |

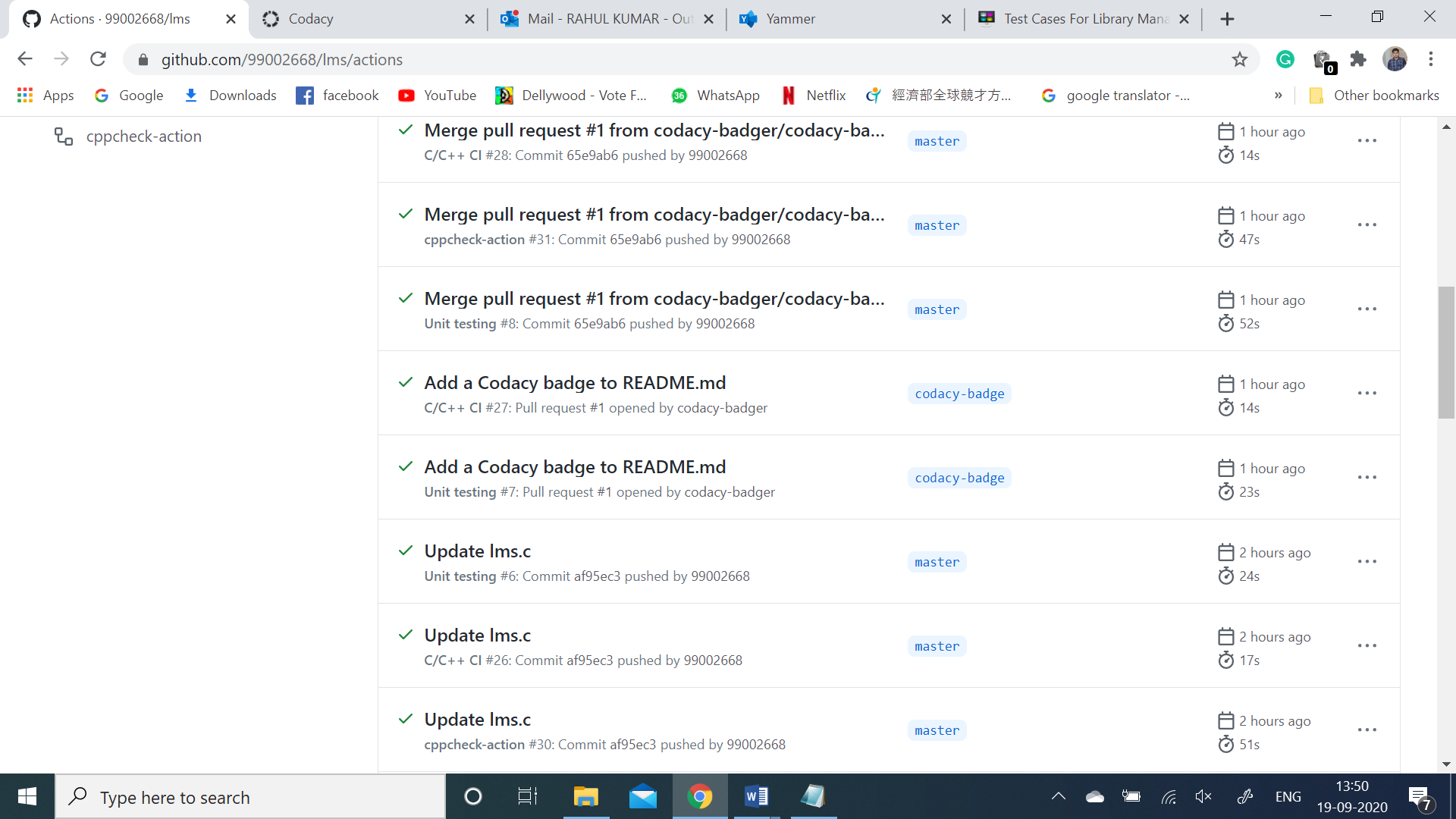
**Low level Test Plan:**

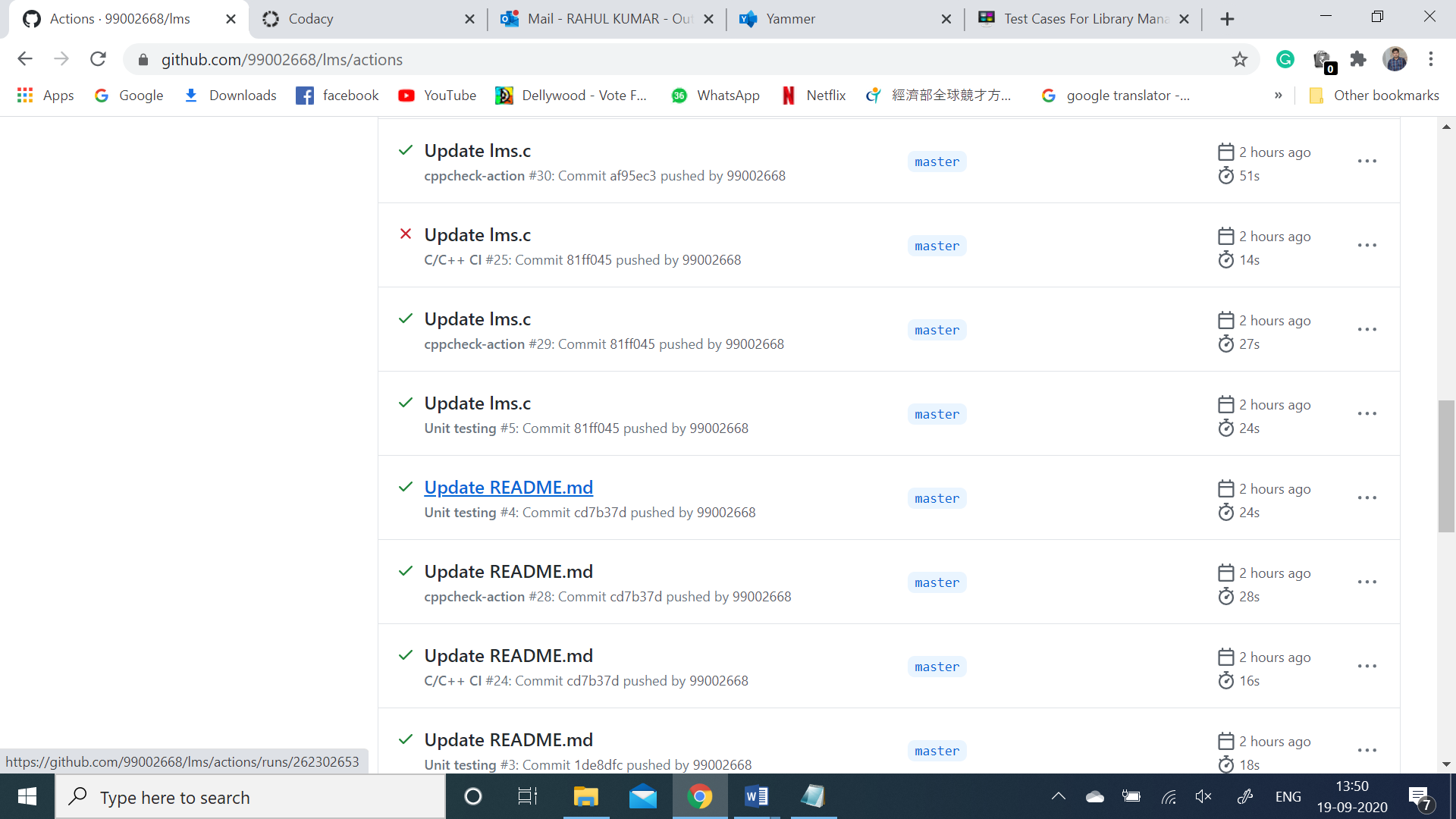
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Precondition** | **Expected Input** | **Expected Output** | **Actual Output** |
| **001** | Enter valid name and password & click on login button | * Software should display main window. | success | success. | success |
| **002** | Enter invaliddetails | Software should not display main window | Successfully show invalid | Successfully show invalid | Successfully show invalid |
| **003** | NEXT  button | Display the next form | Successfully show next step | Successfully show next step | Successfully show next step |
| **004** | CLEAR Button | Display the clear form | Successfully Clear the current record | Added the product to purchase. | Try completing the check-out process by leaving any mandatory field blank in shipping or billing address. |

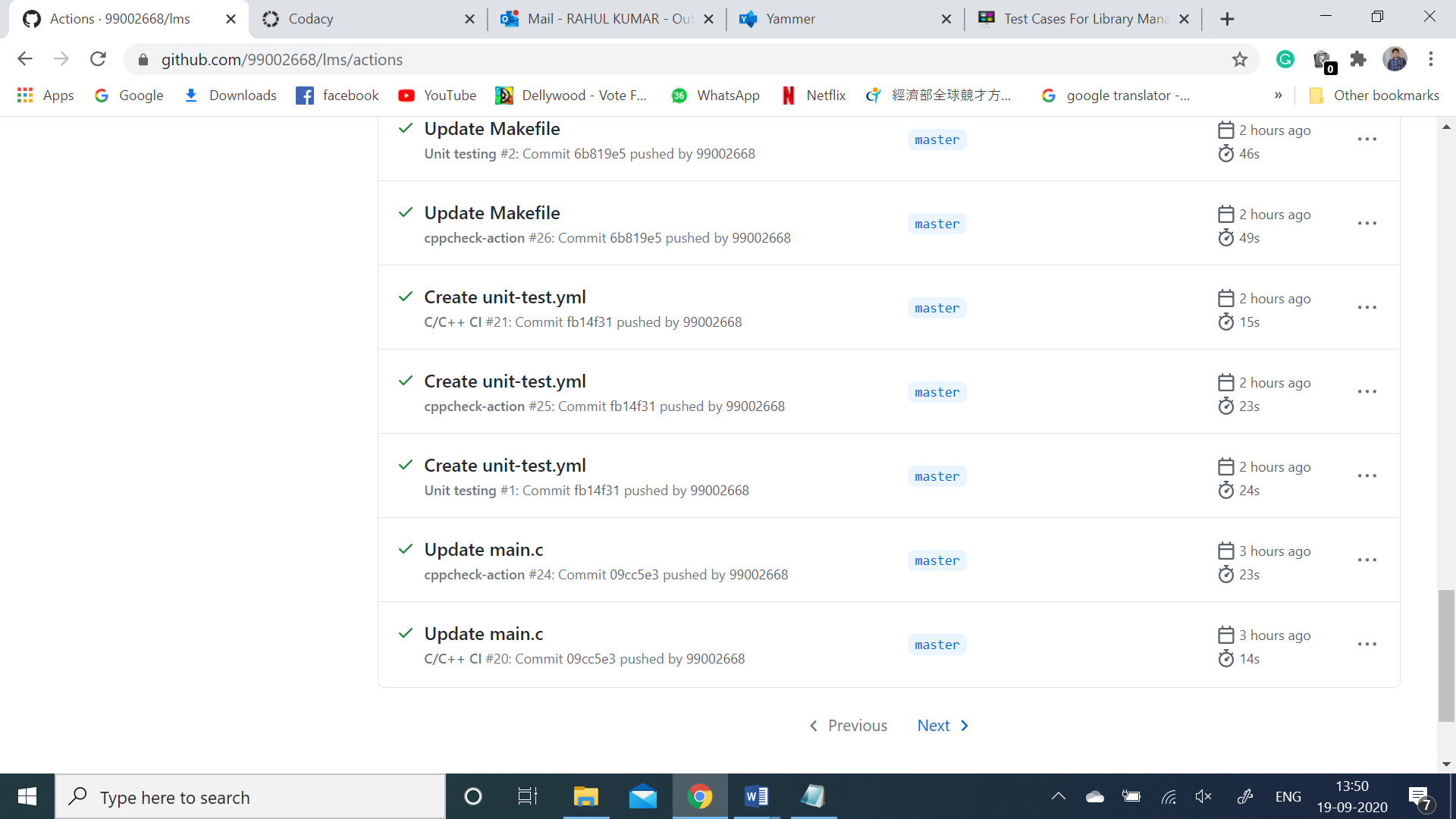
**CI for Applied SDLC and Testing**

[**Github repo**](https://github.com/stepin105343/stepin-practice)

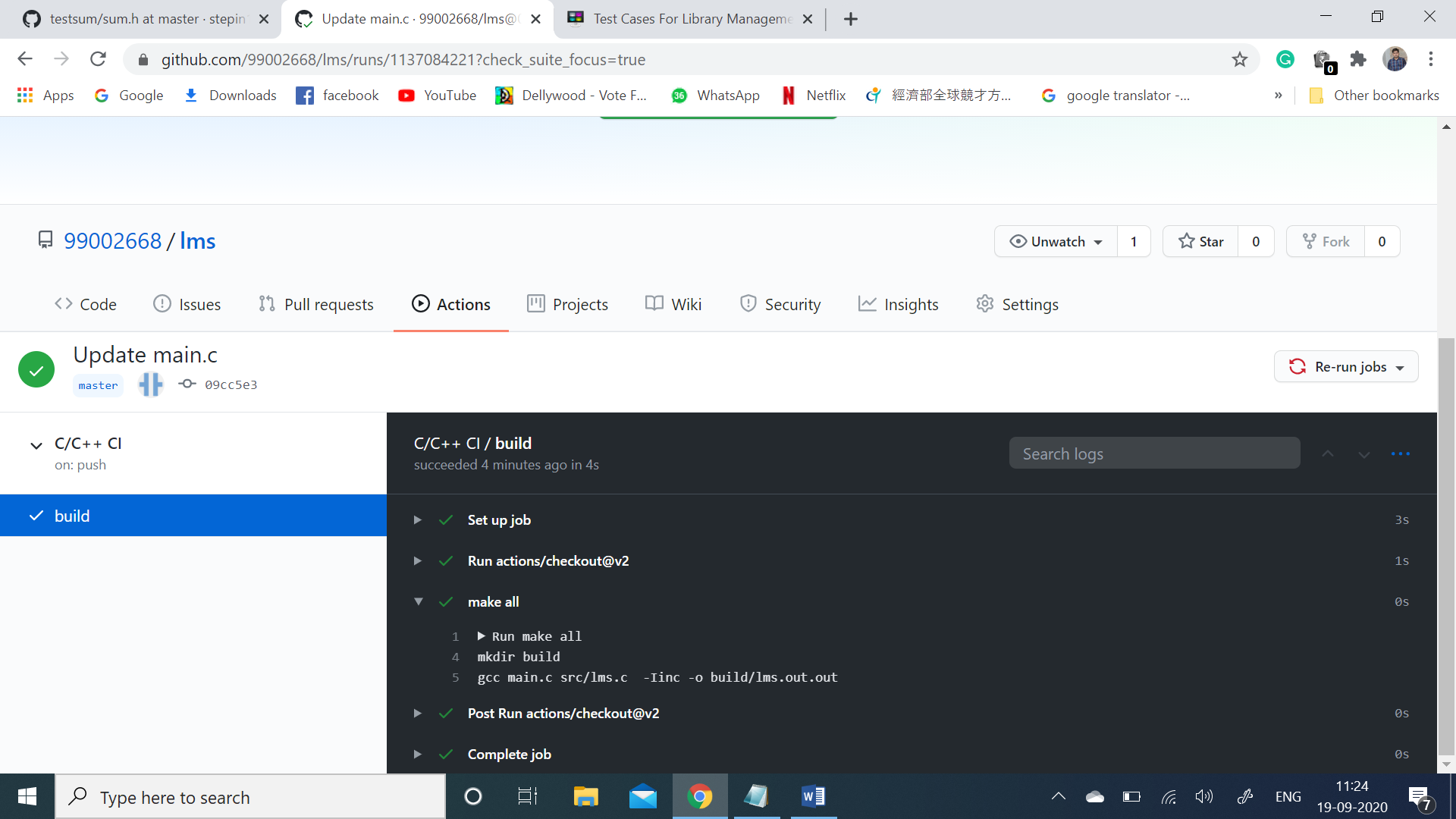
**Actions(Commits):**



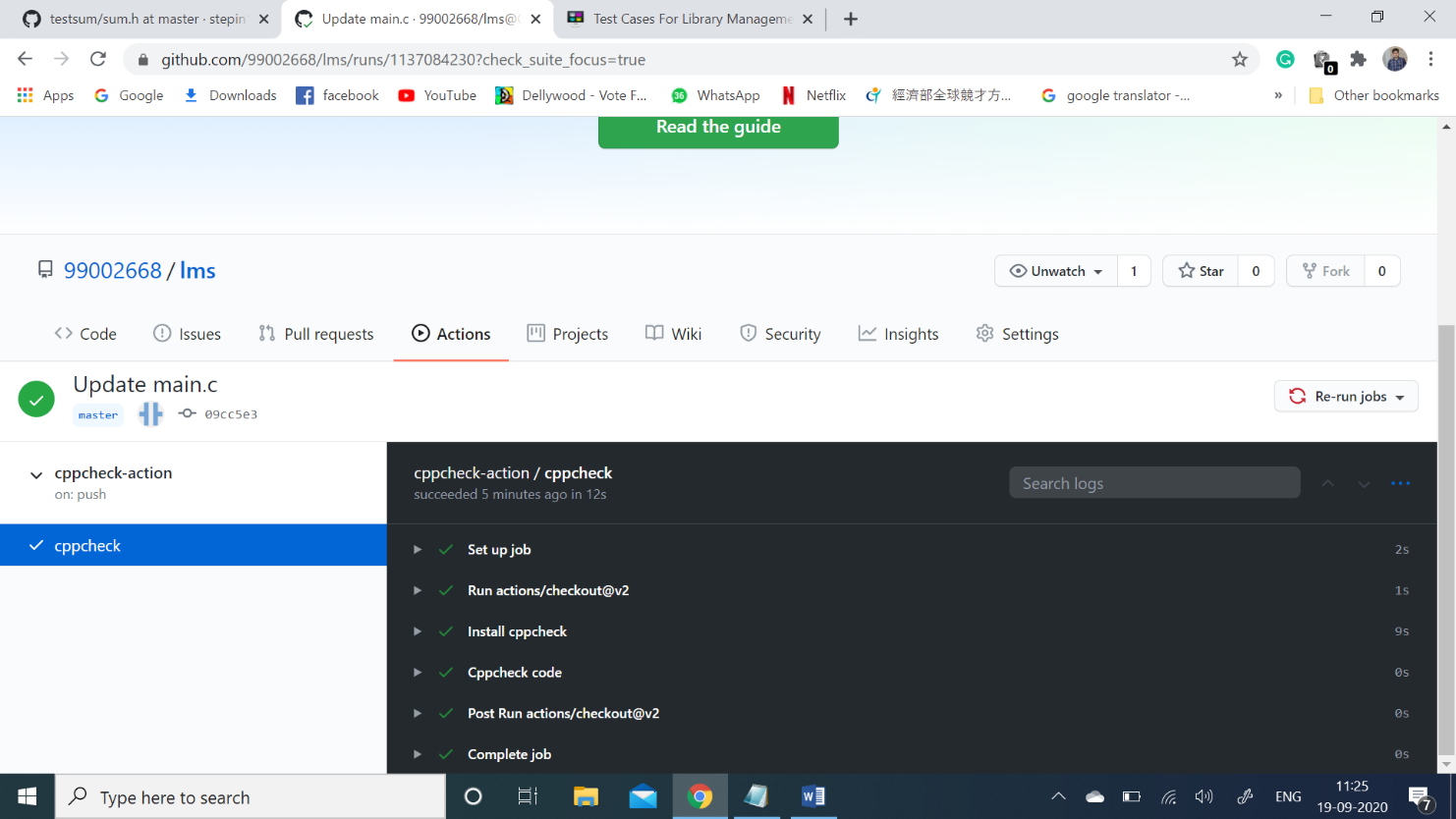




**Build:**



**Cppcheck:**

****

|  |
| --- |
| **Makefile:**  SRC = unity/unity.c\ |
|  |

|  |
| --- |
| src/CheckPrime.c\ |
|  |

|  |
| --- |
| test/Test\_CheckPrime.c\ |
|  |

|  |
| --- |
| main.c |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| INC = -Iunity\ |
|  |

|  |
| --- |
| -Iinc\ |
|  |

|  |
| --- |
| -Itest |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| PROJECT\_NAME = CHECK\_PRIME.out |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| $(PROJECT\_NAME): $(SRC) |
|  |

|  |
| --- |
| gcc $(SRC) $(INC) -o $(PROJECT\_NAME) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| run:$(PROJECT\_NAME) |
|  |

|  |
| --- |
| ./${PROJECT\_NAME} |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| doc: |
|  |

|  |
| --- |
| make -C documentation |
|  |

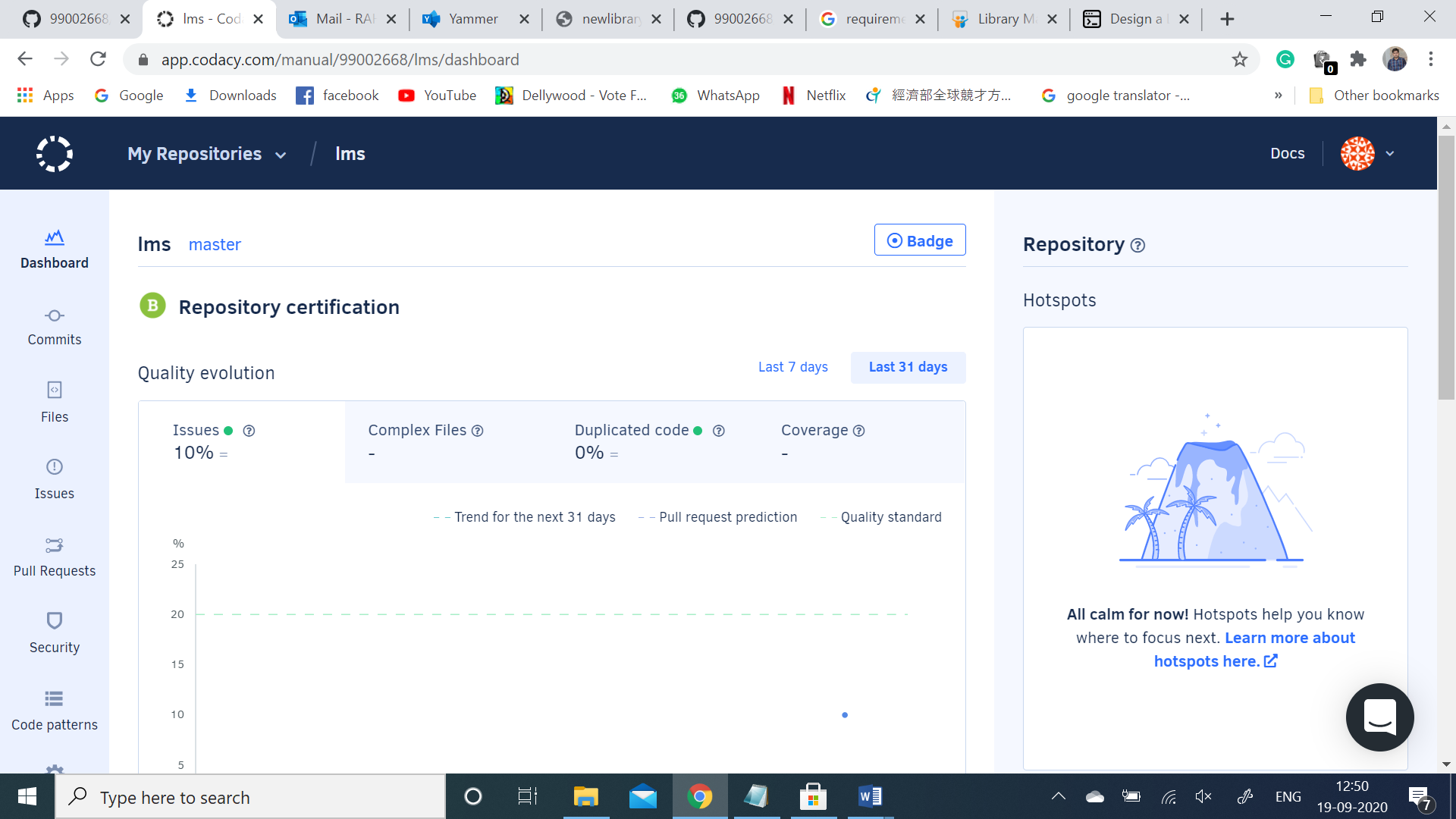
|  |
| --- |
|  |
|  |

|  |
| --- |
| clean: |
|  |

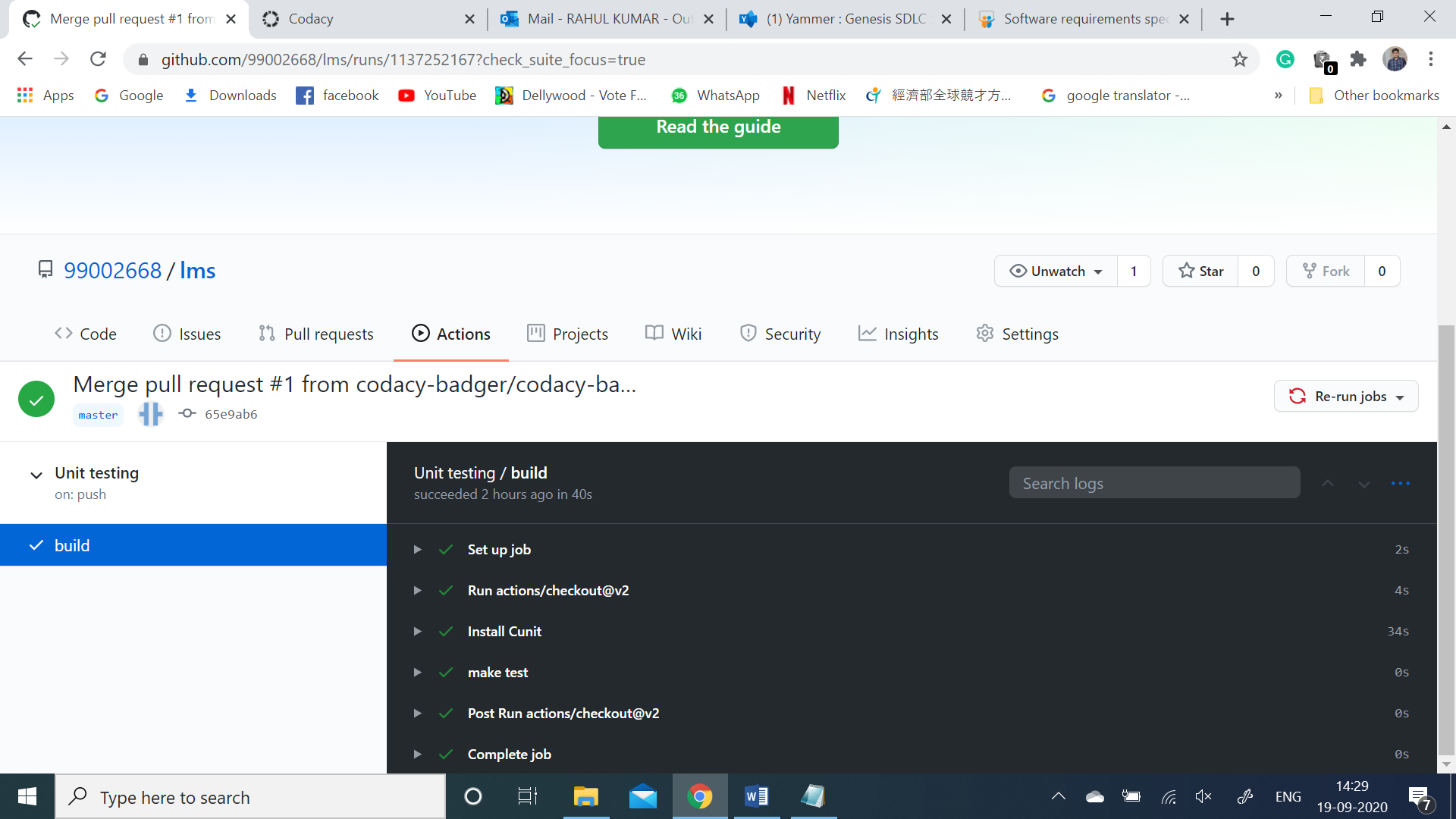
rm -rf $(PROJECT\_NAME) documentation/html

**Code quality**

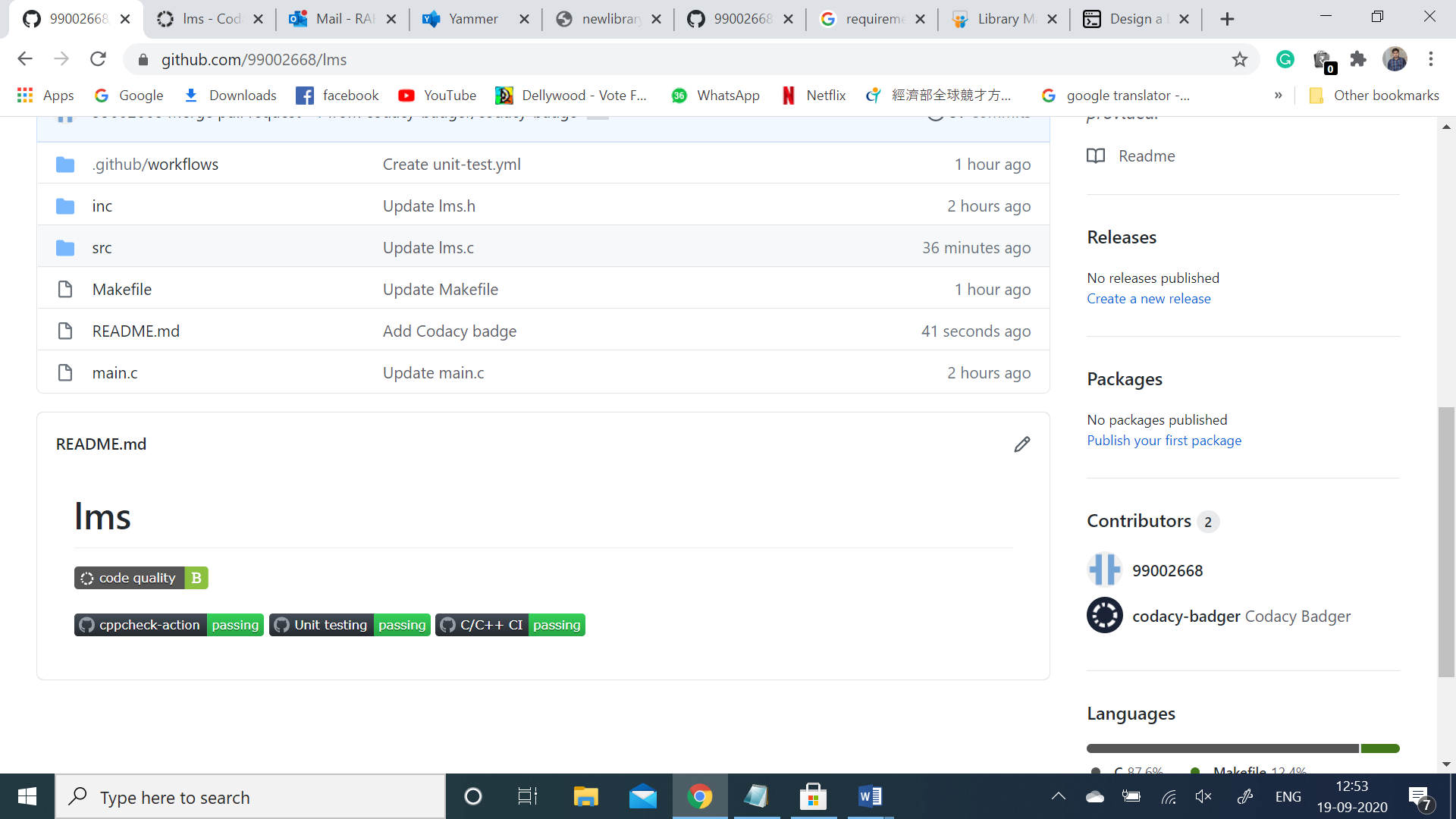
**Codacy details:**



**Unit-Testing:**

****

**Badges for all the above Acticities:**

****

**Appendix:**

<https://github.com/99002668/lms>

**References:**

### <https://github.com/rohan033/Library-Management-System-in-C>

### <https://www.studentprojectguide.com/project-report/test-cases-library-management-system/>

https://www.slideshare.net/Chetan2608/software-requirements-specification-of-library-management-system