|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **Approved By** | **Remarks/Revision Details** |
| **1** | 07/10/2020 | Sachin Kumar |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

# Contents

[Contents 3](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc52942655)

[ACTIVITY 1: ATM MACHINE 4](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc52942679)

[1.Introduction 5](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418128)

[1.1. Research: 5](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418129)

[1.2. Cost Gradation: 6](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418130)

[2.Product Definition: 6](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418131)

[3.SWOT Analysis: 7](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418132)

[4.Requirements: 8](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418133)

[4.1 High level requirements: 8](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418134)

[4.2 Low level requirements: 8](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418135)

[5.System Design 9](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418136)

[Use case Diagram 5](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc52942685)

[Sequence Diagram 6](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc52942686)

[Statechart Diagram 7](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc52942686)

[Activity Diagram 8](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc52942686)

[6.Test Plan 9](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418136)

[Unit Testing 9](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc52942689)

[IntegrationTesting 10](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc52942689)

[7.CI workflow 9](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418136)

[Repository View 12](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc52942689)

[Actions 12](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc52942690)

[Cppcheck 12](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc52942691)

[Code Quality 13](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc52942695)

[5. System Design 9](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418136)

[5.1 Data Flow Diagram: 9](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418137)

[5.2 Sequence Diagram: 9](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418138)

[5.3 Activity Diagram: 10](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418139)

[5.4 Class Diagram: 11](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418140)

[6. Appendix 26](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418157)

[Activity 2 - Calculator 18](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418148)

[1. Introduction 18](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418149)

[2.Requirements 19](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418150)

[2.1 High Level Requirements: 19](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418151)

[2.2 Low Level Requirements: 20](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418152)

[3.System design 21](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418153)

[3.1 Use case Diagram 21](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418154)

[3.2 Activity Diagram 22](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418155)

[3.3 Flowchart Diagram 23](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418155)

[4.Test Plan 24](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418156)

[5.CI Workflow 25](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418156)

[6.Appendix 26](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418157)

[Activity 3 – Agile Development model 27](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418148)

[1. Introduction 27](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418149)

[1.1 theme: 27](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418151)

[1.2 initiative: 27](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418151)

[1.3 user stories: 28](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418151)

[2.Appendix 28](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418157)

[Activity 4 – cREATE MAKEFILE 29](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418148)

[1. Introduction 29](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418149)

[2. Appendix 29](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc51418157)

[References 30](file:///C:\Users\sachi\OneDrive\desktop\LR_AppliedSDLC_99002462.docx#_Toc52942704)

# 

# Activity 1 – ATM MACHINE

**GitHub Repository Link -** [**Link**](https://github.com/99002462/ATM-Machine)

## **Introduction**

The ATM stands for Automated Teller Machines (ATM), which is an interface of the work done in banks in electronic format. An automated teller machine (ATM) is computerized telecommunications device that provides a financial institution's customers a secure method of performing financial transactions, in a public space without the need for a human bank teller. Through ATM, customers interact with a user-friendly interface that enables them to access their bank accounts and perform various transactions. The atm-machine, which I have designed, will have,

* Features like deposit, withdraw, check balance, transfer money and exit button.
* In transfer money, section the user will be able to transfer money between its own accounts(i.e. Savings to Current Account and Current to Savings Account)
* Deposit and Withdraw balance operations will show the left balance after every transaction.

## **Research:**

**1.1 Ageing**

* The idea of out-of-hours cash distribution developed from bankers' needs in Japan, Sweden, the United Kingdom, and the United States.
* A Japanese device called the "Computer Loan Machine" supplied cash as a three-month loan at 5% p.a. after inserting a credit card. The device was operational in 1966
* The roll-out of this machine, called Bankograph, was delayed by a couple of years, due in part to Simjian's Reflectone Electronics Inc. being acquired by Universal Match Corporation
* The first ATM was set up in June 1967 on a street in Enfield, London at a branch of Barclays Bank, credited to a British inventor named John Shepherd-Barron.
* This instance of the invention is credited to the engineering team led by John Shepherd-Barron of printing firm De La Rue, who was awarded an OBE in the 2005 New Year Honours
* ATMs can be placed at any location but are most often placed near or inside banks, shopping centers/malls, airports, railway stations, metro stations, grocery stores, petrol/gas stations, restaurants, and other locations. ATMs are also found on cruise ships and on some US Navy ships, where sailors can draw out their pay.

## **1.2. Cost Analysis:**

**How much does it cost a bank to establish an ATM?**

* Cost of ATM machine - ₹ 2,50,000/-
* Interiors, 2ACs & Cameras - ₹ 1,00,000/-
* Uninterrupted Power System - ₹ 1,30,000/-

**Cost of establishing an ATM *- ₹ 4, 80,000/-***

**How much does it cost a bank to maintain an ATM?**

* Electricity Bill - ₹ 12,000/- per machine
* 24x7 security & Maintenance - ₹ 30,000/-
* Rent - ₹ 5,000/- (market dependent)

**Cost of maintaining an ATM *- ₹ 47,000/- per month***

**NOTE:**

* ATMs have 24x7 service as against 8 hrs in branch.
* ATMs have more reachability than a branch.
* ATMs are faster and easier to maintain.

|  |  |
| --- | --- |
| Best ATM Models | Price |
| 1. Genmega 2500, Onyx | 1. $2,550 |
| 1. Hantle 1700 | 1. $2175 |
| 1. Nautilus Hyosung Halo | 1. $1,940 |
| 1. Triton Argo, Traverse | 1. $1930 |

# 2. Product Definition:

The proposed wireless headphones should have:

* 24\*7 hrs service: The ATM works on electricity so it works until the electricity supply is not interrupted.
* Transfer Funds: Transfer funds between your Checking, Savings and even a Line of Credit if it is associated with your deposit account.
* Fast Cash: Use this convenient feature to withdraw a quick ₹400 from your Savings Account. (₹1000 is the default amount on all ATMs, but this can be updated in the ATM preferences to whatever amount works for you.)
* Balance Inquiry: ATM should be able to display the information, ledger balance and current balance in the account.

# 3. SWOT Analysis:

1. Advantages of Automated Teller Machine:

* The ATM provides 24 hours service.
* The ATM provides privacy in banking communications.
* The ATMs reduce the workload banks staff.
* The ATM may give customer new currency notes.
* The ATMs are convenient for banks customers.
* The ATM is very beneficial for travellers.
* The ATM provides services without any error.

1. Features of Automated Teller Machine:

* Transfer funds between linked bank accounts
* Receive account balance
* Prints recent transactions list
* Change your pin
* Deposit your cash
* Prepaid mobile recharge
* Bill payments
* Cash withdrawal

1. Disadvantages of ATM Machines

* ATM usage fees is usually high
* Lack of personal service and safety

1. Opportunities:

* Increased revenue: Revenue could increase because of the commission you earn on each transaction as well as from new customers who are entering your establishment to use the ATM.
* Reduced credit card processing fees: Businesses with ATM machines located inside their shop receive more cash payments compared to businesses that do not. By putting an ATM machine inside your store, you can drive down credit card processing fees.
* Convenience: Providing your customers with an easy cash option inside your business means you are providing them with a convenience.
* Flexible plans: ATM machine companies often help with marketing and will allow you to buy or lease your machine, depending on what works best for your business. There are short-term contract and cancel-anytime options, and some companies help you with marketing the ATM.

1. Threats:

* Portable technology is evolving every day.
* Backhoes
* Weak Physical Locks.

## **Requirements**

### **High Level Requirements:**

|  |  |
| --- | --- |
| **ID** | **Description** |
| HL\_01 | An atm machine application that should perform basic operations to deposit, withdraw, check balance and transfer money. |
| HL\_02 | The application should be developed using standard C++ language and should run on all machines supporting G++ compiler. |
| HL\_03 | Should display following menu to users like 1. Deposit, 2. Withdraw, 3. Check Balance, 4. Transfer Money, 5. Exit. |

Table 1: High Level Requirements for ATM Machine

### **Low Level Requirements:**

|  |  |
| --- | --- |
| **ID** | **Description** |
| LL\_01 | Should exit when entered anything else than the options provided. |
| LL\_02 | Prevent users from changing any data regarding to their account. |
| LL\_03 | User should be able to avail one functionality at one time. |
| LL\_04 | Application should terminate if wrong pin is entered. |
| LL\_05 | User should be able to check balance and transfer amount in both savings and current account. |
| LL\_06 | User should be able to make a balance enquiry of any account. |

Table 2: Low Level Requirements for ATM machine

## **System design**

### Use case Diagram

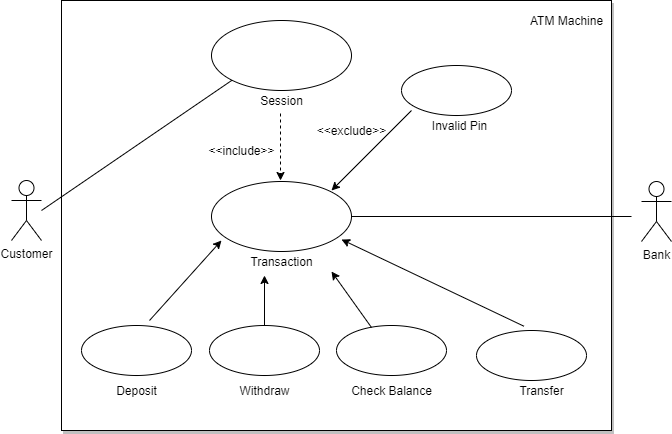


Figure 1: Use Case Diagram for ATM machine

### Sequence Diagram

Figure 2: Sequence Diagram for ATM machine

1. **State-chart Diagram**

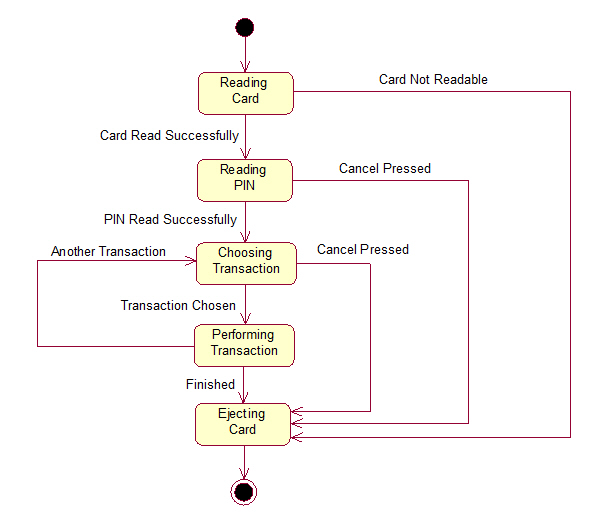


Figure 3: State-Chart Diagram for ATM machine

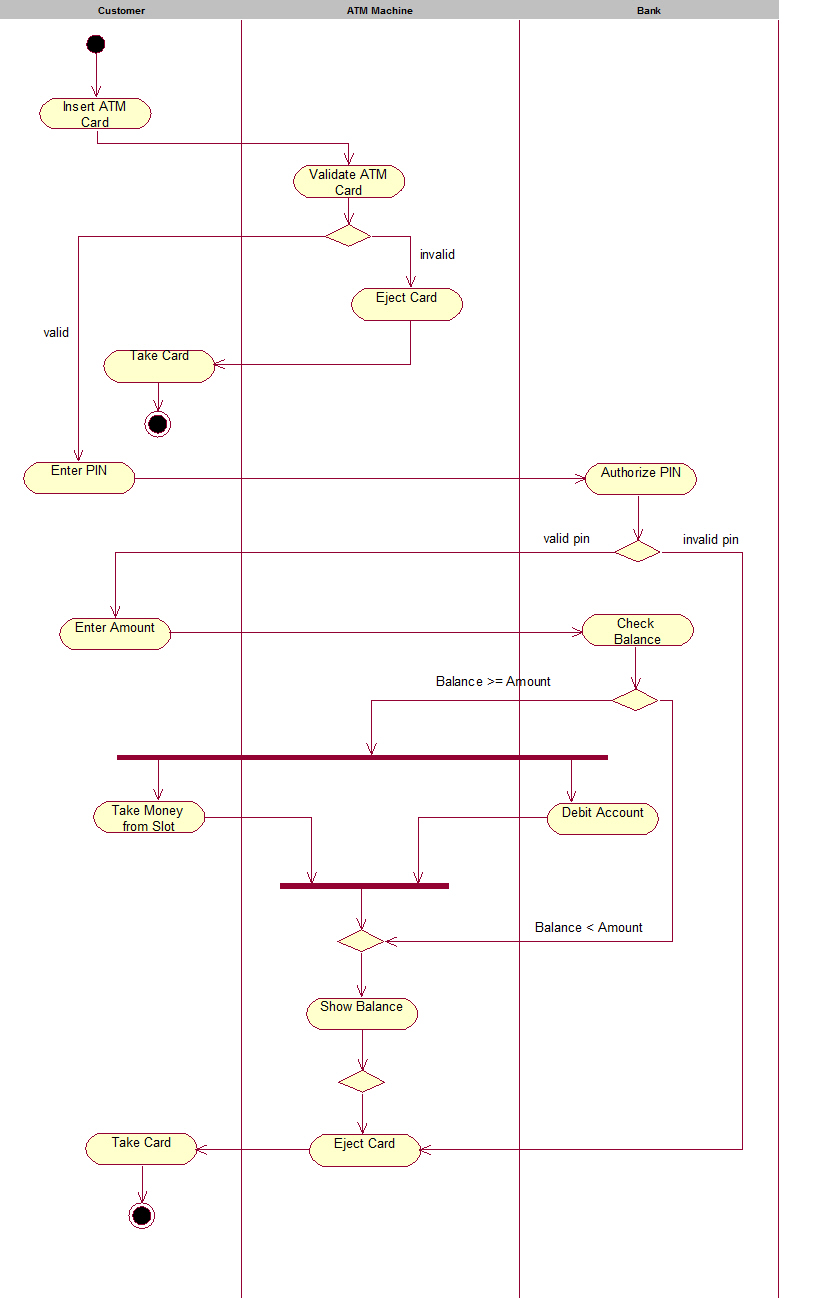
1. **Activity Diagram**

Figure 4: Activity Diagram for ATM machine

## **Test Plan**

1. **Unit Testing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Description | Pre-Condition | Expected input | Expected output | Actual output |
| T\_01 | Successful card insertion. | Card should be approved by using valid PIN in the positive integer format | 1234 | Welcome “Sachin Kumar” |  |
| T\_02 | Verify that the user is presented with options when the card is inserted correctly | Card should be verified by the pin and it should be of 4 integer | 1234 | Display the main menu options |  |
| T\_03 | Verify that if a total number of incorrect pin attempts gets surpassed then the user is not allowed to continue further and operations will be terminated | Options should be from the main menu | 1,2,3,4,5 | Display the menu according to the selected options |  |
| T\_04 | Verify that the user is presented with different account type options like- saving and current | Click on the 1st,2nd,3rd options from the menu | 1,2,3 | 1. Current 2. Savings |  |
| T\_05 | Verify the cash deposit functionality. | User need to choose type of bank | 100 | Display the message:  “Your updated Savings balance is: $100” |  |
| T\_06 | Verify the cash withdrawal functionality by inserting some valid amount. | Should be positive integer value and user must have added some amount before withdrawing in the same instance | 10 | Display the message:  “Your updated balance is: $90” |  |
| T\_07 | Enquire the balance | User need to access the account using valid pin | 1,2 | Display the message:   1. Your current Checking balance is $0 2. Your current savings balance is $100 |  |
| T\_08 | User able to transfer the amount | User have to choose the option from savings to current or current to savings from the menu | 1,2 | Display the message:  “Enter transfer amount” |  |
| T\_08 | User able to exit from the instance | User should be able to login using valid PIN | 5 | Thank you for Choosing our Bank! Stay Safe! Stay Healthy! |  |

Table 3: Unit Test Plan for ATM machine

1. **Integration Testing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Description | Pre-Condition | Expected input | Expected output | Actual output |
| T\_01 | Verify that check balance functionality | User should be logged into the account PIN and choose the type of account | 3 | Display the message:  “Your current Savings balance is $0” |  |
| T\_02 | Verify the transfer functionality | User should have sufficient balance in the savings and current account | 4 | Display the main options:   1. Current to Savings 2. Savings to Current |  |

Table 4: Integration Test Plan for ATM machine

## **CI Workflow**

### **Repository View -** [Link](https://github.com/99002462/ATM-Machine)

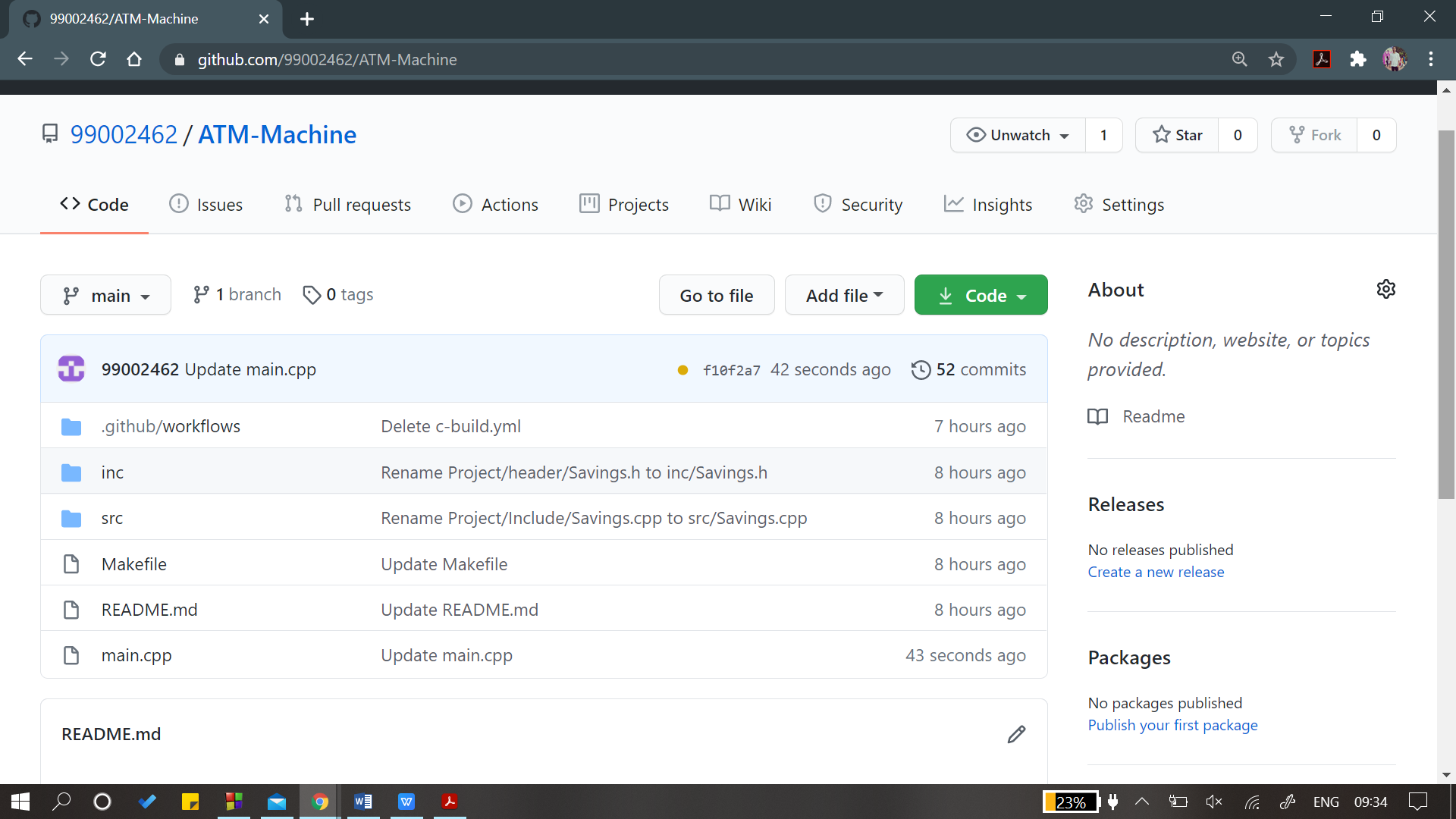


Figure 5: Repository View

### 

### **Actions:** [**Link**](https://github.com/99002462/ATM-Machine/actions)

Figure 6: Actions

### **Cppcheck:** [**Link**](https://github.com/99002462/ATM-Machine/actions?query=workflow%3Acppcheck-action)

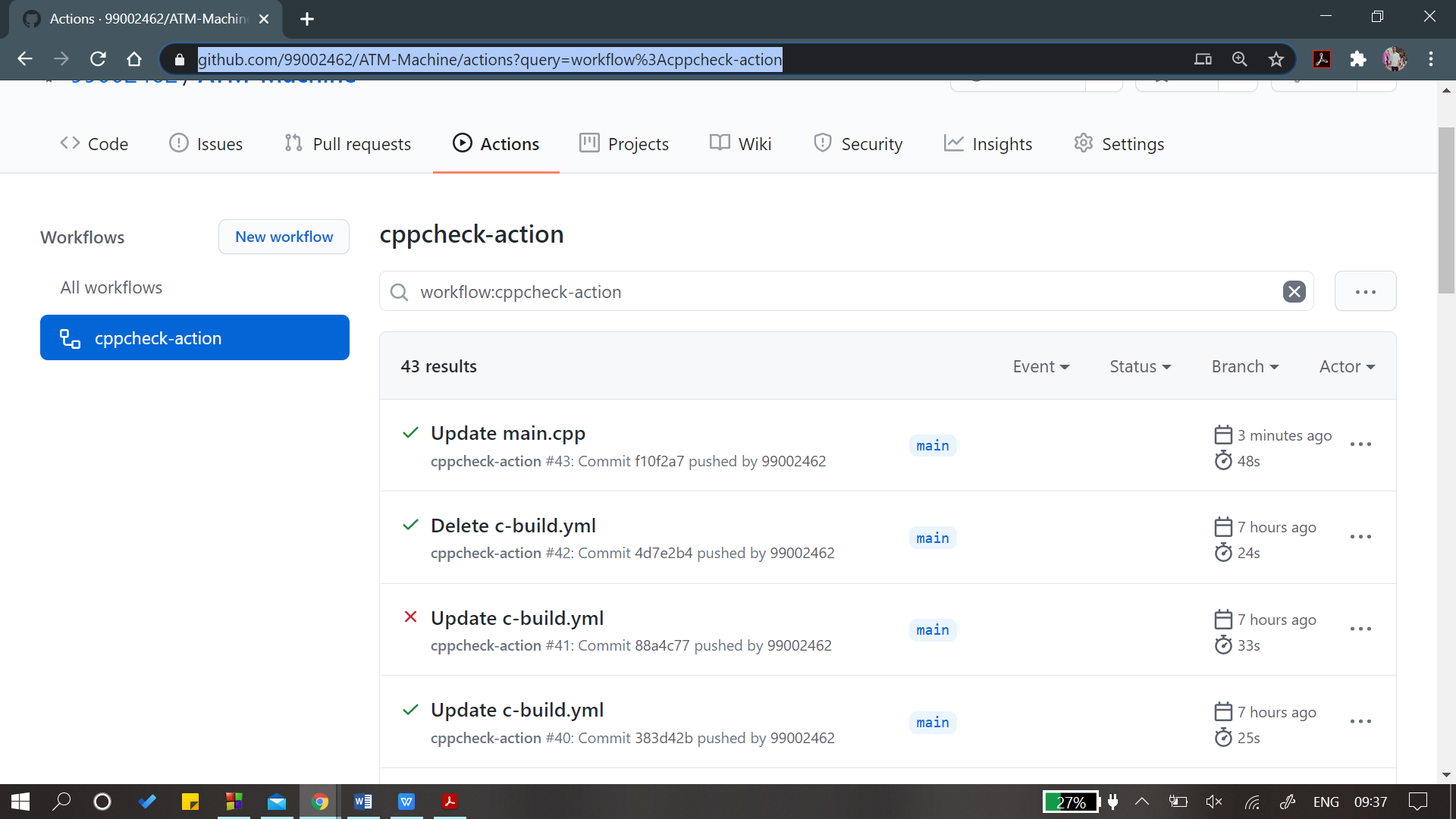


Figure 7: Cppcheck

### **Code Quality :** [**Link**](https://app.codacy.com/organizations/gh/99002462/dashboard)

Figure 8: Code Quality

# Appendix

### **Repository View -** [Link](https://github.com/99002462/ATM-Machine)

* + 1. **The work that I liked:-**

1. In this activity, I liked working on research, ageing and cost analysis.
   * 1. **Challenges I faced:-**
2. I faced issue in working on the documentation and making the test plans.
3. Consolidating the documentation.

# Activity 2 – Calculator

# 1. Introduction

A calculator app is one of the most basic yet important apps on your phone. You need to deal with calculations every day and a calculator app allows you to use your smartphone for all the calculations on the go. Even though the default calculator app in most Android smartphones today is fairly feature packed and capable enough to handle a few complex equations. If you need a scientific calculator that can help you with all types of calculations and math problems, you will have to go for a third-party calculator app.

The calculator we have designed will have,

* Simple Calculations like addition, subtraction, multiplication, division and modulo division.
* Scientific Operations like nth power of a number, square root of a given number, factorial of a number and multiplicative inverse of a number.
* Conversion operations like currency conversion (US Dollars to Indian Rupees), Length conversion (Feet to Inches) and time conversion (Hours to Minutes).

# 2. Requirements

**2.1. SWOT Analysis**

**Strengths**

• Environmentally friendly application.

• Can perform operations anytime and every time.

**Weaknesses**

• Does not perform all operations.

• There may be some incorrect results due to the invalid or wrong user input.

**Opportunities**

• Saves human power to calculate manually.

• Innovative technology.

**Threats**

• When some invalid user input is given, it may end the program.

## **2.1 High Level Requirements:**

|  |  |
| --- | --- |
| ID | Description |
| HL01 | A mobile calculator app that should perform simple calculations, scientific calculations and conversions. |
| HL02 | The calculator was developed using standard C language and should run on all machines supporting gcc compiler. |
| HL03 | Should display following menu bar to users like-1.Add, 2.Subtract, 3.Multiply, 4.Divide, 5.Modulus, 6.Power, 7.Square root, 8.Factorial, 9.Inverse, 10.Currency,  11.Length, 12.Time, 13.Exit |

## **2.2 Low Level Requirements:**

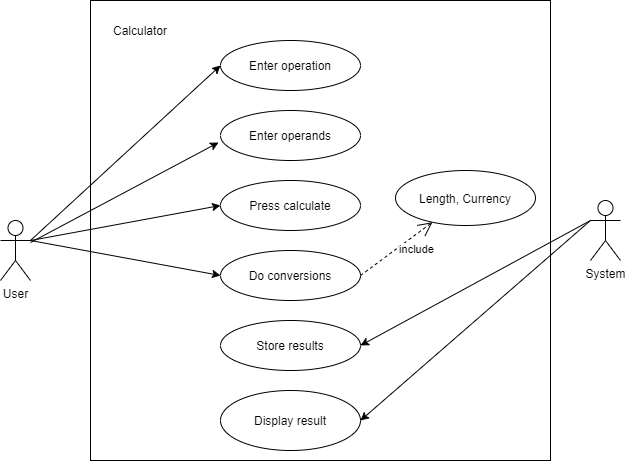
|  |  |
| --- | --- |
| ID | Description |
| L01 | Should exit when entered 13 |
| L02 | Prevent users from divide by zero error. |
| L03 | Can use either one or two operands |
| L04 | Should display “Invalid Selection” when user chooses menu option less than zero and greater than 13. |

# 

# 

# 3. System design

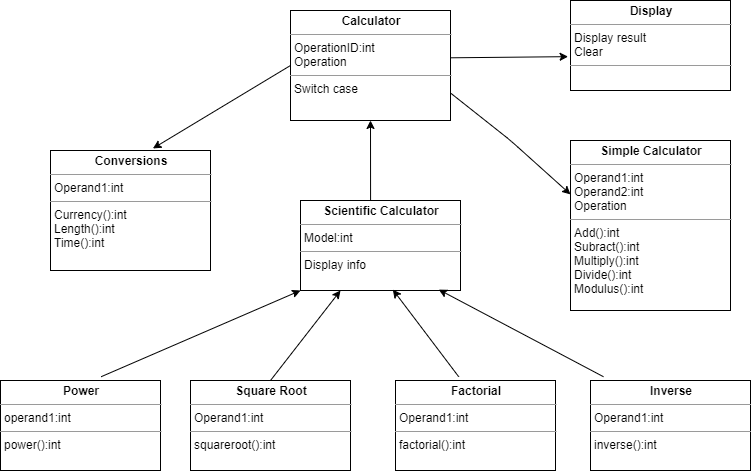
## **3.1 Use case Diagram**



**Fig.5 Use Case Diagram for calculator**

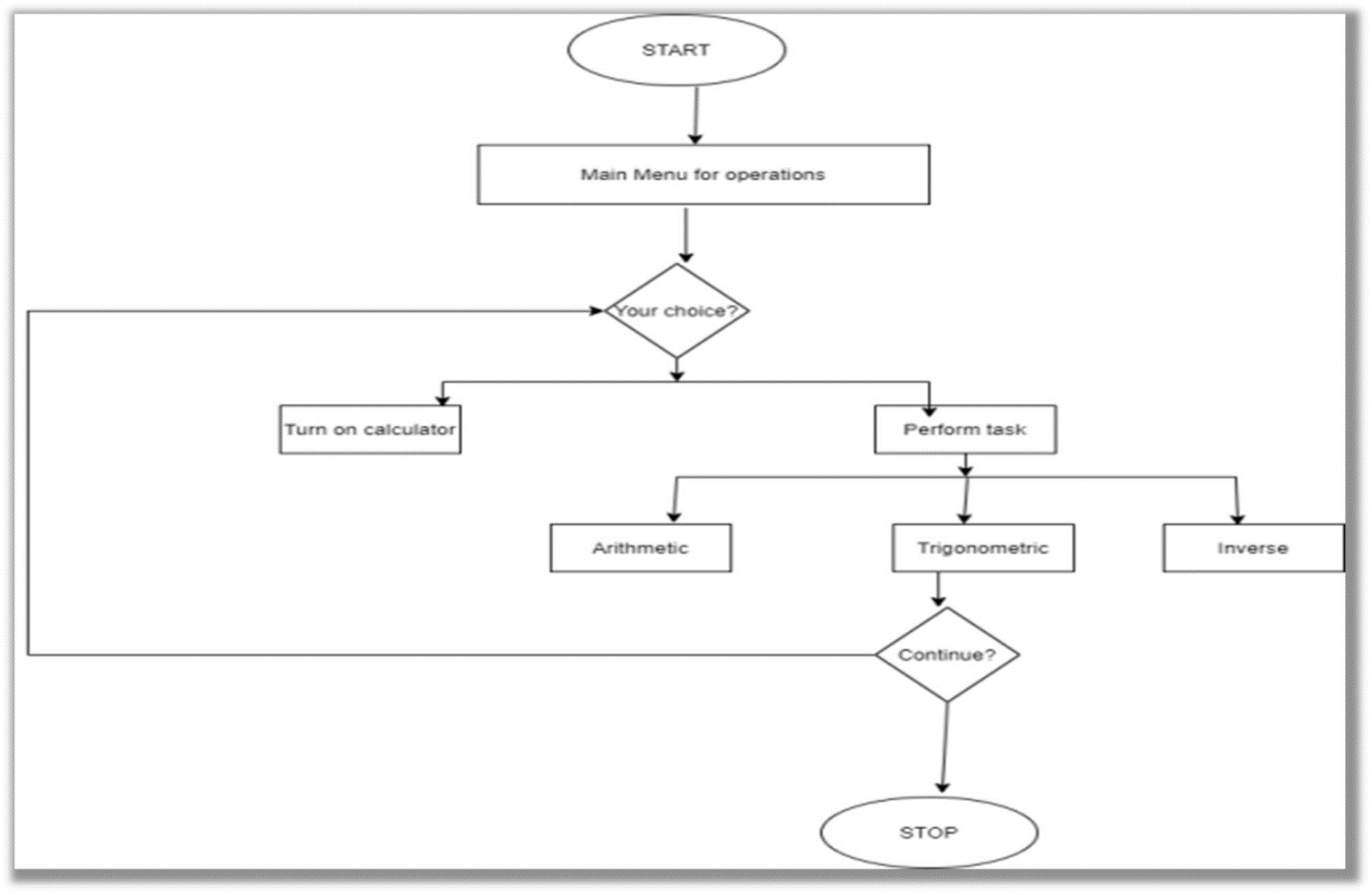
## 

## **3.2 Activity Diagram**



**Fig.6 Activity Diagram for Calculator**

## **3.2 Flowchart Diagram**



**Fig.7 Flowchart Diagram for Calculator**

**4. Test Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Description | Pre-Condition | Expected input | Expected output | Actual output |
| Test\_01 | Work For function add | Operands must be integers | 7+26 | 33 |  |
| Test\_02 | Work For function add | Operands must be integers | 234578+345698 | 580276 |  |
| Test\_03 | Work for function multiply | Operands must be integers | 2\*4 | 8 |  |
| Test\_04 | Work for function multiply | Operands must be integers | 2456\*3456 | 8487936 |  |
| Test\_05 | Work for function divide | Operands must be integers | 7/0 | Divide by zero error |  |
| Test\_06 | Work for function square root | Operand must be integer | 25 | 5 |  |
| Test\_07 | Work for function square root | Operand must be integer | 8 | 2.82 |  |
| Test\_08 | Work for function  currency | Operand must be integer | 2 | 148 |  |
| Test\_09 | Work for function  currency | Operand must be integer | 2.5 | Error |  |
| Test\_10 | Work for function  currency | Operand must be integer | 2,5 | Error |  |

# CI Workflow

**5.1.** **Repository View:** [Link](https://github.com/99002462/Calculator)

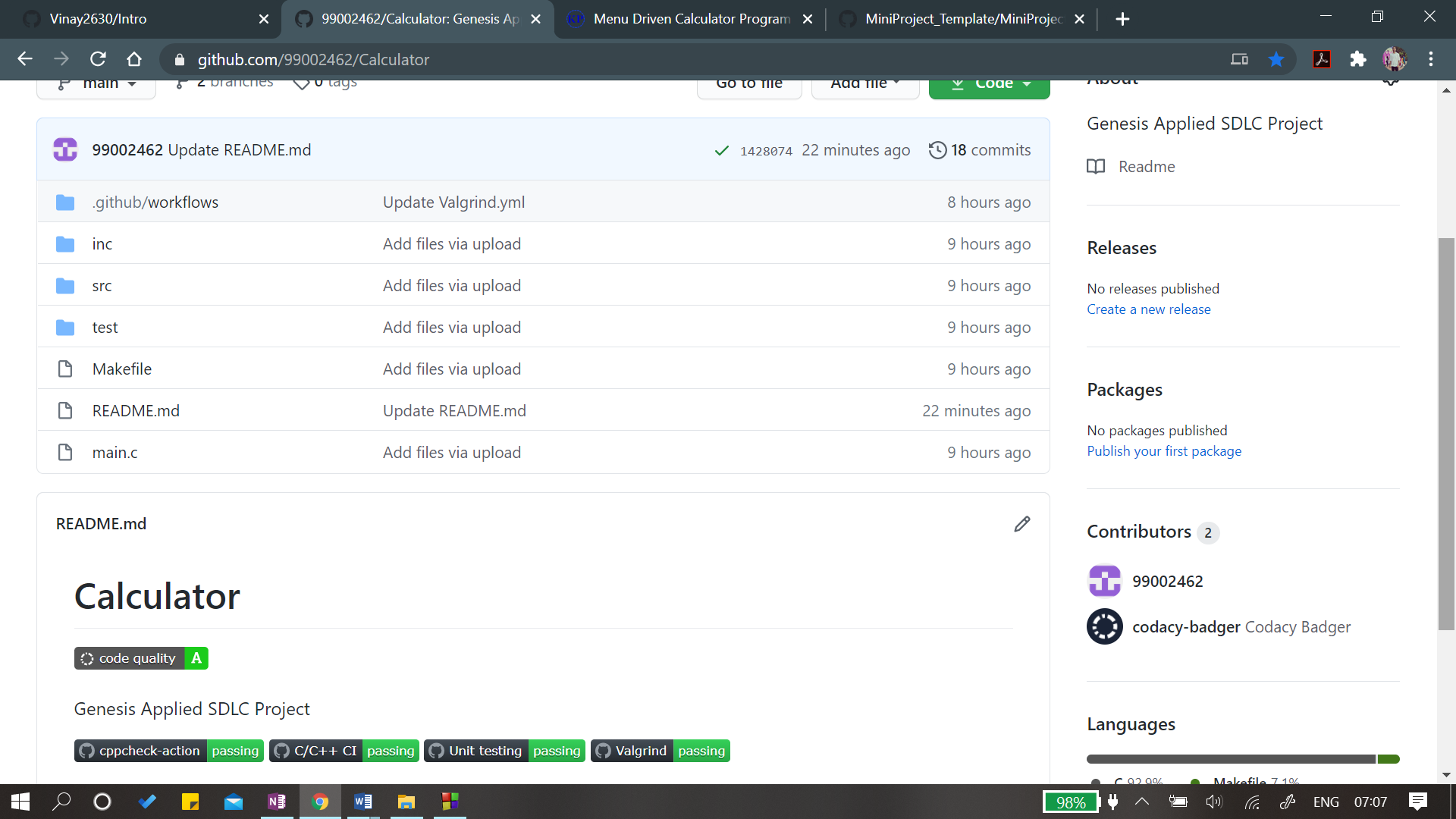


Figure 5: Repository View

### **5.2. Actions:** [**Link**](https://github.com/99002462/Calculator/actions)

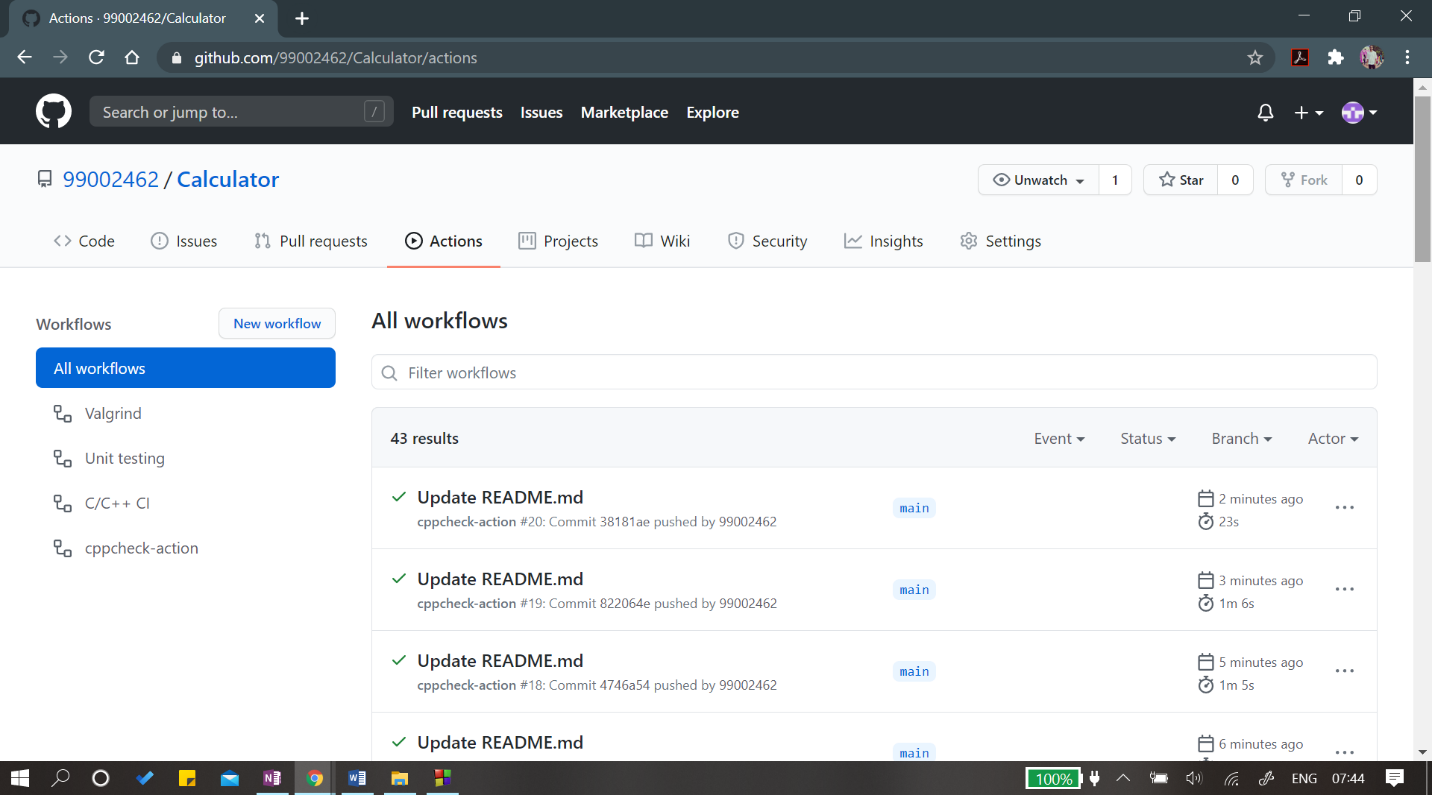


Figure 6: Actions

### **5.3. Cppcheck:** [**Link**](https://github.com/99002462/Calculator/actions?query=workflow%3Acppcheck-action)

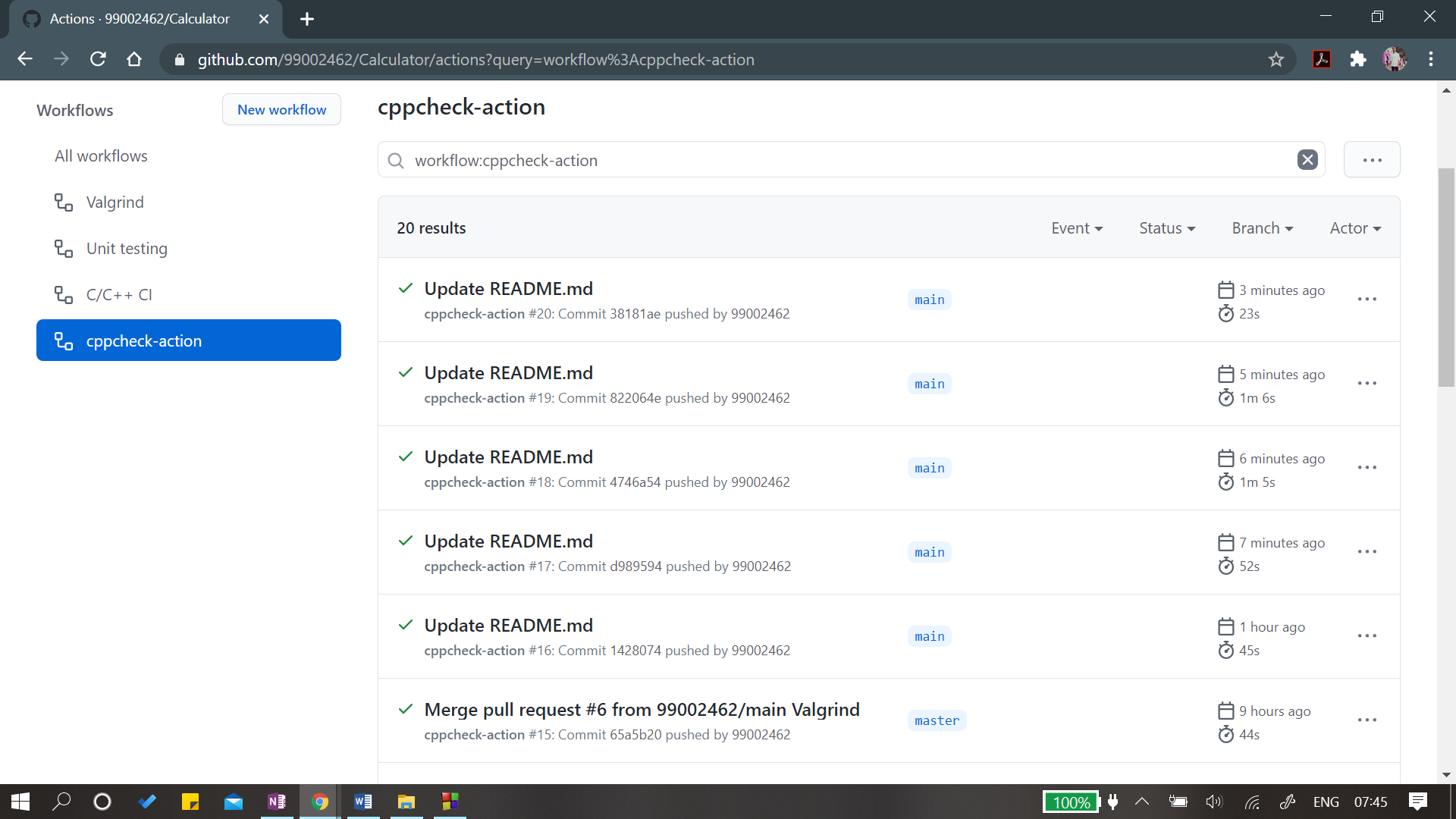


Figure 7: Cppcheck

### **5.4. Code Quality:** [**Link**](https://app.codacy.com/gh/99002462/Calculator/dashboard)

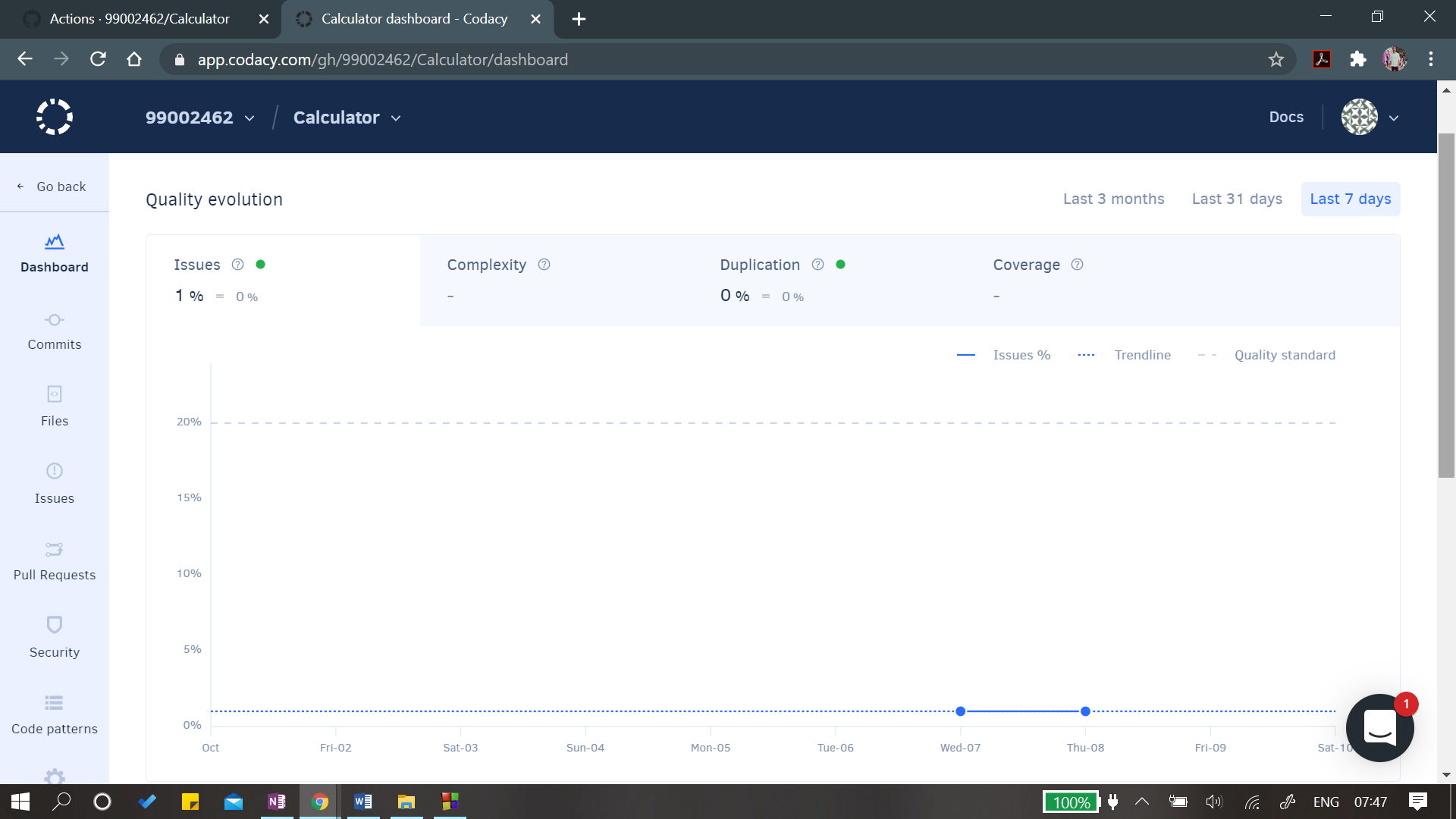


Figure 8: Code Quality

**5.5. C/C++ CI-Build:** [**Link**](https://github.com/99002462/Calculator/actions?query=workflow%3A%22C%2FC%2B%2B+CI%22)

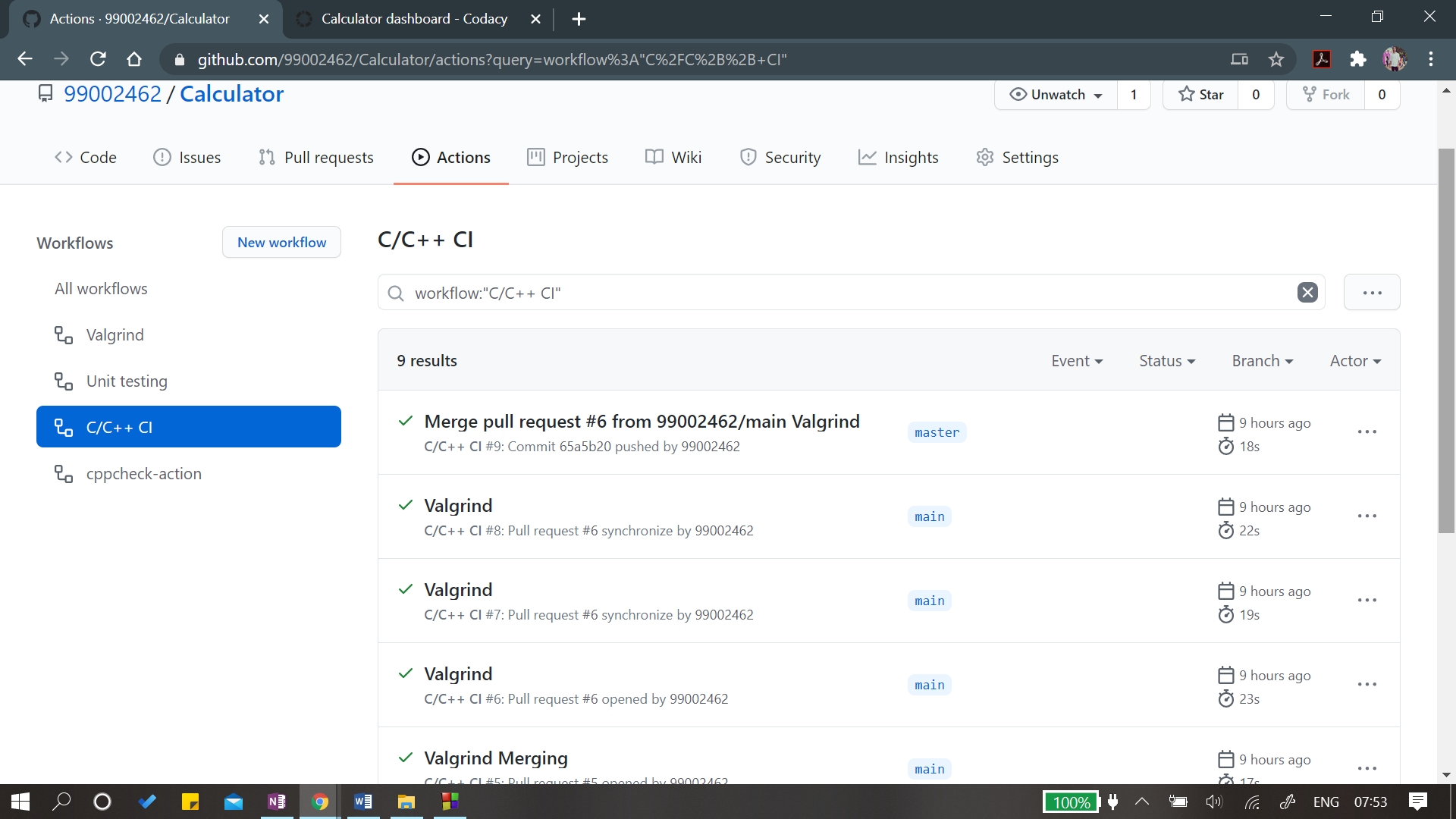


Figure 8: C/C++ CI Build

**5.5. Unit Testing :** [**Link**](https://github.com/99002462/Calculator/actions?query=workflow%3A%22Unit+testing%22)

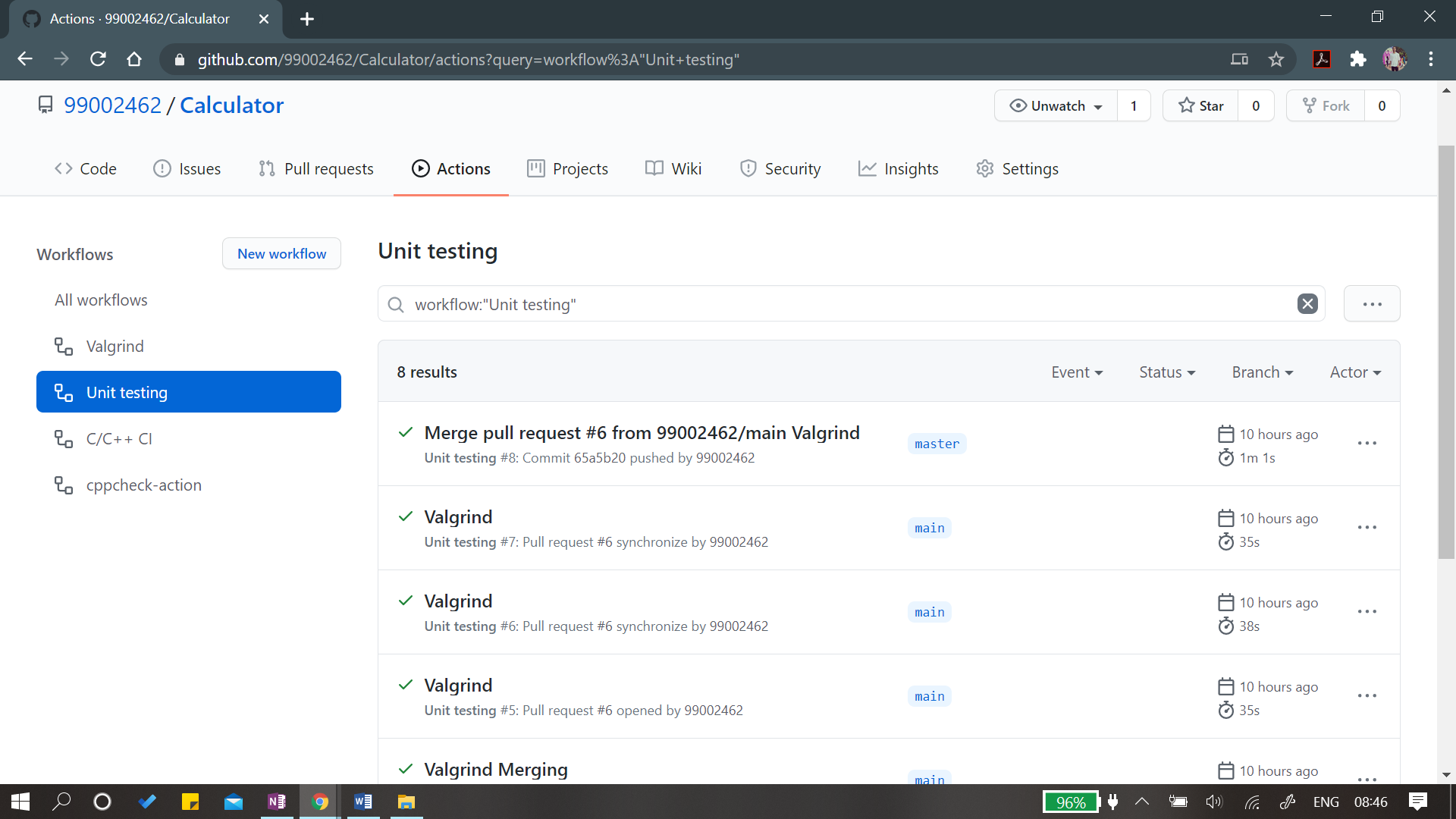


Figure 9: Unit Testing

**5.5. Valgrind:** [**Link**](https://github.com/99002462/Calculator/actions?query=workflow%3AValgrind)

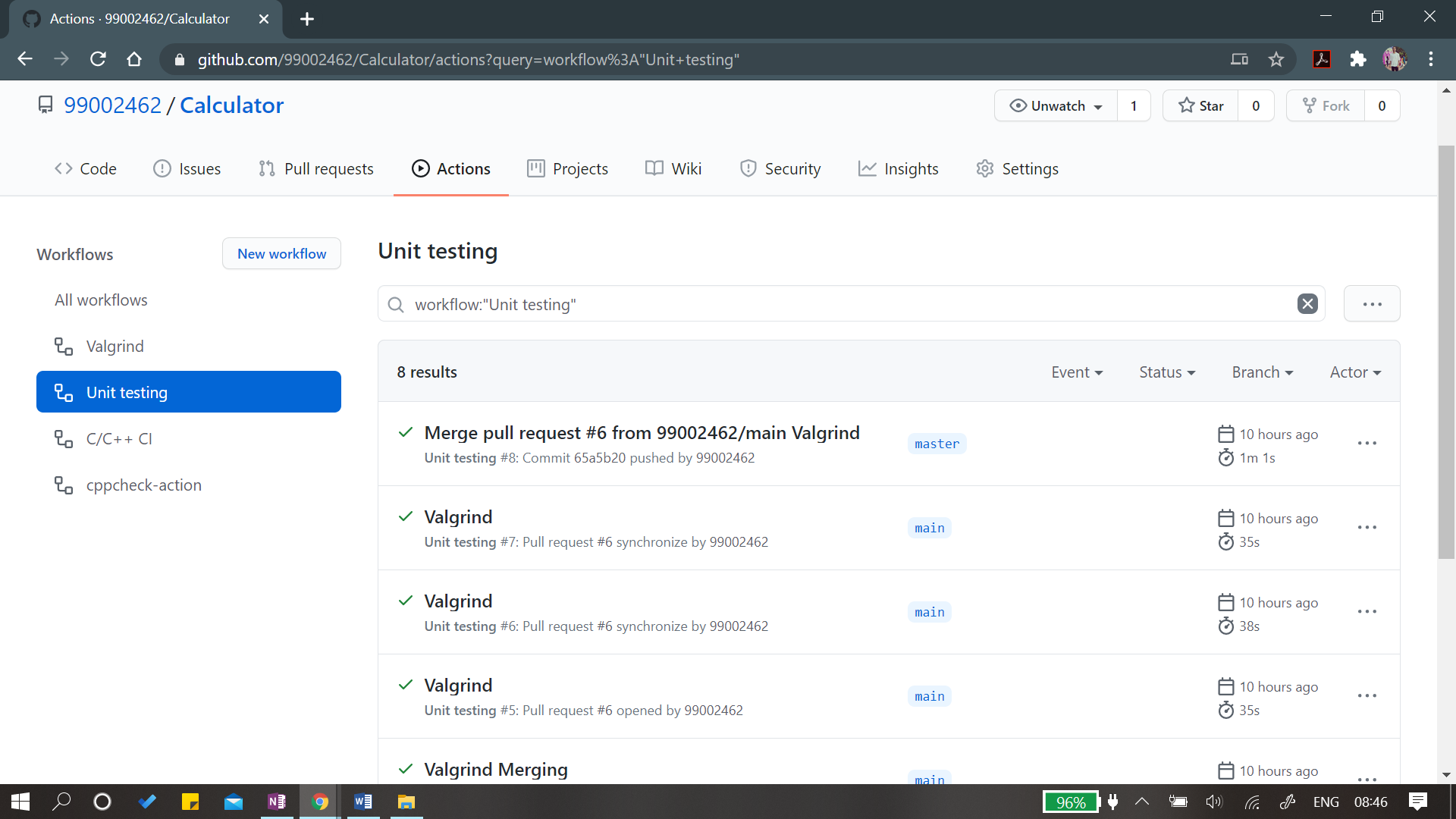


Figure 10: Valgrind

# Appendix

**GitHub Link:** [**Link**](https://github.com/99002462/Calculator)

* + 1. **The work that I liked:-**

1. I like to debug the errors while building the project in GitHub and I have learned the meaning of many errors and where to look for clearing those errors**.**
   * 1. **Challenges I faced:-**
2. I faced issue in obtaining the Valgrind badge, so I learned about valgrind testing and how it is done on Linux, and then I overcame this obstacle.

**Activity 3 – Agile Development Model**

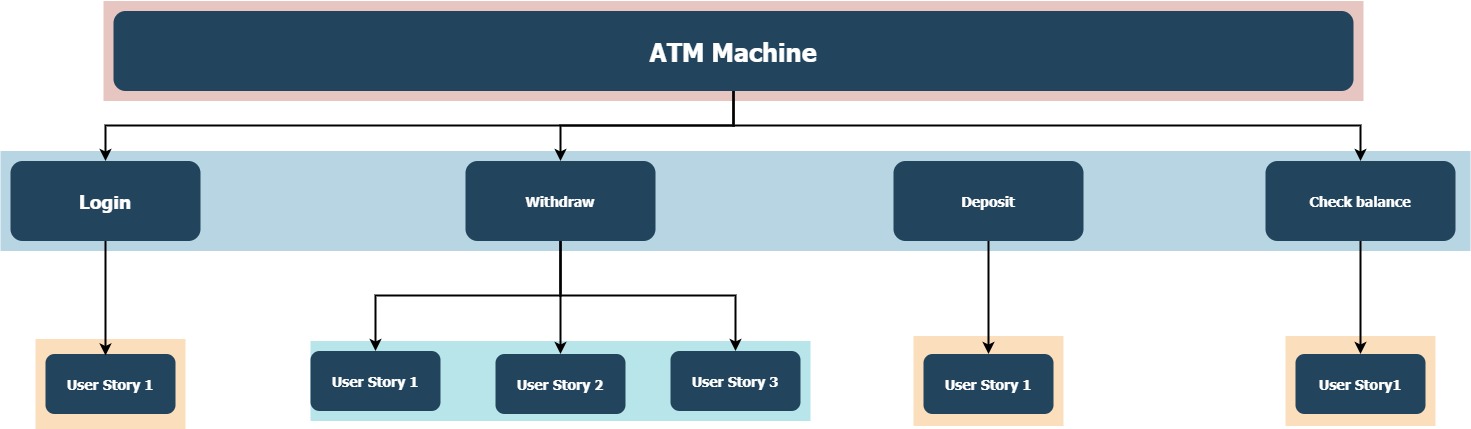
1. **INTRODUCTION**

Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer issues to complete the project before deadline. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments. Requirements, plans, and results are evaluated continuously so that teams have a natural mechanism for responding to change quickly.

## **THEME: ATM Machine**

I have used Scrum Methodology for Activity 1 (i.e. ATM Machine) to develop my product. This helps to set the expectations with stakeholders and other teams also.

**Initiative:**



**User Stories:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | |  |  |  |
| **ID** | **Epic** | **User Story** | **As a/an** | **Acceptance Criteria** | **Notes** | **Story Point**  **(Days)** |
| 1 | Withdraw Cash | Select Account | ATM User | User will be presented with a list of their accounts to select from.Accounts displayed will indicate account type (i.e. savings, checking)User will be presented error message if the user selects any other option | Checking account will be displayed | 1 |
| 2 | Withdraw Cash | Enter Amount | ATM User | User will be able to enter the amount of the withdrawal.User may not enter $0 to withdraw.User will be presented with an error message if they do not enter any numeric valueUser will be able to cancel the transaction.User will be able to change the amount entered. | Display only in dollarsConsider to track the account chosen for transactionBalance to check balance. | 1 |
| 3 | Withdraw Cash | Validate Funds | ATM User | User is not able to withdraw more money than the account balance.User will be presented with an insufficient funds message if the amount entered to withdraw is more than the account balance.If the user is presented with an insufficient funds message, they have to insert less amount then available balance. | Error message will be highlighted in red.Allow navigation to account balance.Obtain error message from business. | 2 |
| 4 | Withdraw Cash | Withdraw Amount | ATM User | User will be presented the amount of the withdrawal.User will be presented the account for the withdrawal. | Shows message for the amount, which is chosen to withdraw.  * Shows message the left amount after withdrawal | 3 |
| 5 | Login | Enter Pin | ATM user | System must validate the card and pin codeIn case Customer enters wrong Pin code three times then the system locks the card. | If Pin is correct it should show the main menuIf Pin is wrong then the instance should be closed | 4 |

# Appendix

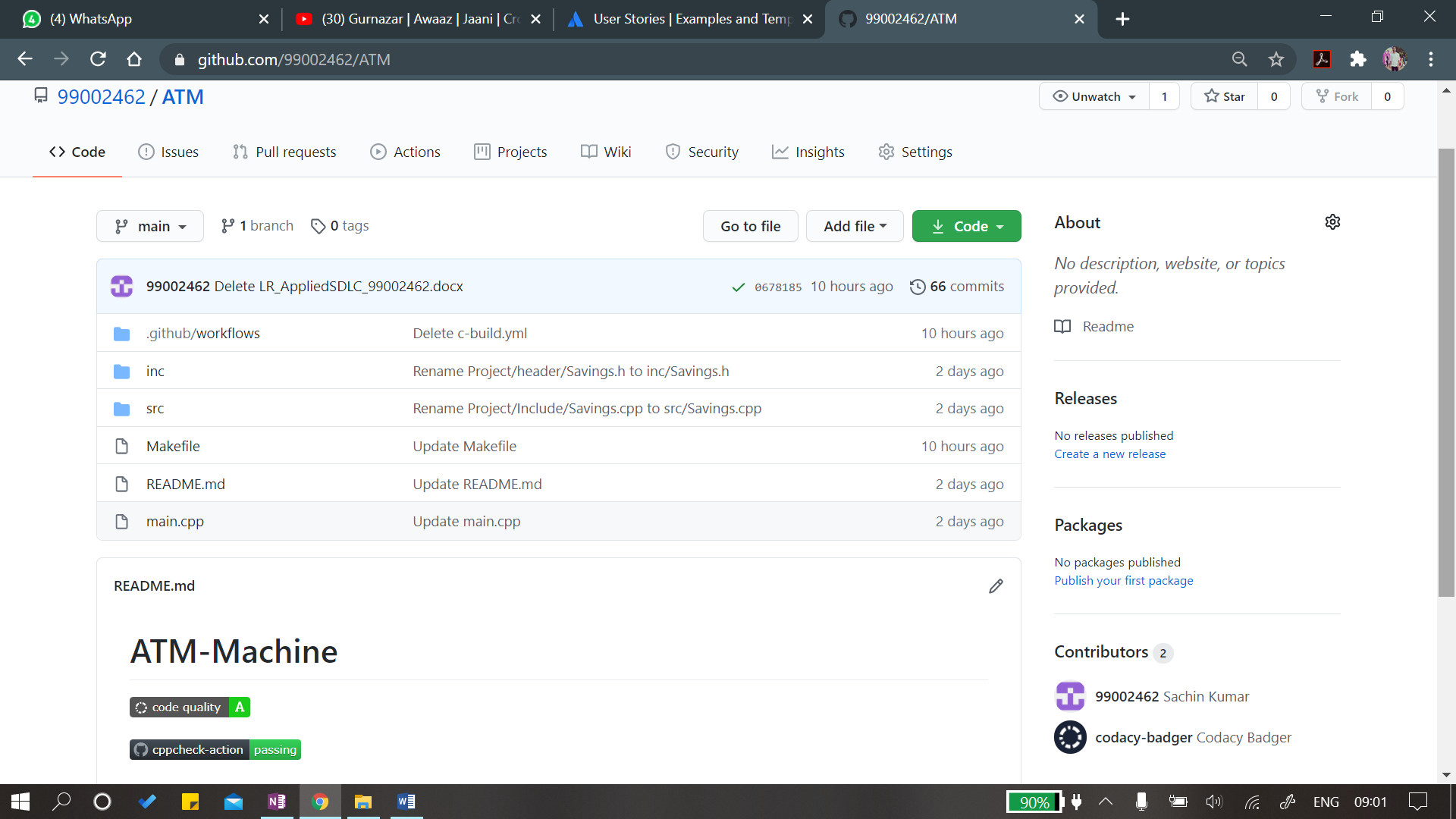
* 1. **The work that I liked:-**

1. Creating the User Stories after discussing a questionnaire with my peers
   1. **Challenges I faced:-**
2. Writing a good user stories so that it can help during testing.

**Activity 4 – Create Make File**

**Introduction:**

A makefile is a file (by default named "Makefile") containing a set of directives which is used to build automation tool to generate a target/goal. This makefile helps to build the multiple files automatically.



# Appendix

**GitHub Link** : [Link](https://github.com/99002462/ATM-Machine)

* 1. **The work that I liked:-**

1. After completion of the code, I like to make the workflow for the project and take badges.
   1. **Challenges I faced:-**
2. Debugging the source code so that makefile can run smoothly.

# References

1. <https://www.knowprogram.com/c-programming>
2. <https://draw.io/>
3. <https://github.com/stepin654321/MiniProject_Template>
4. <https://www.startertutorials.com/uml/>
5. [https://github.com/Vinay2630/Intro](https://github.com/Vinay2630/Calculator)
6. <https://www.businessnewsdaily.com/>
7. <https://www.quora.com>
8. <https://www.flashcardmachine.com/scrum-user-stories.html>