|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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](#_Toc52942655)

[Table of Figures: 3](#_Toc52942656)

[Table of Tables: 3](#_Toc52942657)

[ACTIVITY 1: ATM MACHINE 4](#_Toc52942679)

[Introduction 4](#_Toc52942680)

[Requirements 4](#_Toc52942681)

[High Level Requirements: 4](#_Toc52942682)

[Low Level Requirements: 5](#_Toc52942683)

[System design 5](#_Toc52942684)

[Use case Diagram 5](#_Toc52942685)

[Sequence Diagram 6](#_Toc52942686)

[Statechart Diagram 7](#_Toc52942686)

[Activity Diagram 8](#_Toc52942686)

[Test Plan 9](#_Toc52942687)

[Unit Testing 9](#_Toc52942689)

[IntegrationTesting 10](#_Toc52942689)

[CI Workflow 11](#_Toc52942688)

[Repository View 12](#_Toc52942689)

[Actions 12](#_Toc52942690)

[Cppcheck 12](#_Toc52942691)

[Code Quality 13](#_Toc52942695)

[References 13](#_Toc52942704)

# Table of Figures:

[Figure 1: Use Case Diagram for ATM Machine 5](#_Toc52136269)

[Figure 2: Sequence Diagram for ATM Machine 6](#_Toc52136270)

[Figure 3: Statechart Diagram for ATM Machine 7](#_Toc52136270)

[Figure 4: Activity Diagram for ATM Machine 8](#_Toc52136270)

[Figure 5: Repository View 11](#_Toc52136273)

[Figure 6: Actions 12](#_Toc52136274)

[Figure 7: Cppcheck 12](#_Toc52136275)

[Figure 8: Code Quality 13](#_Toc52136279)

# Table of Tables:

[Table 1: High Level Requirements for ATM machine 4](#_Toc52136502)

[Table 2: Low Level Requirements for ATM machine 4](#_Toc52136503)

[Table 3: Unit Test Plan for ATM machine 9](#_Toc52136504)

[Table 4: Integration Test Plan for ATM machine 10](#_Toc52136504)

# 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.

## **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

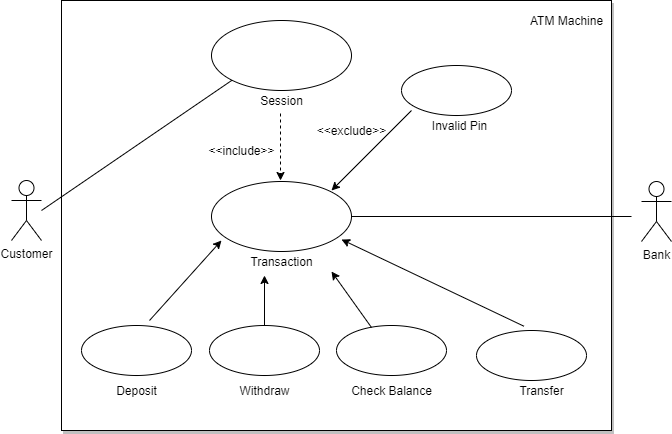


Figure 1: Use Case Diagram for ATM machine

### Sequence Diagram

Figure 2: Sequence Diagram for ATM machine

1. **State-chart Diagram**

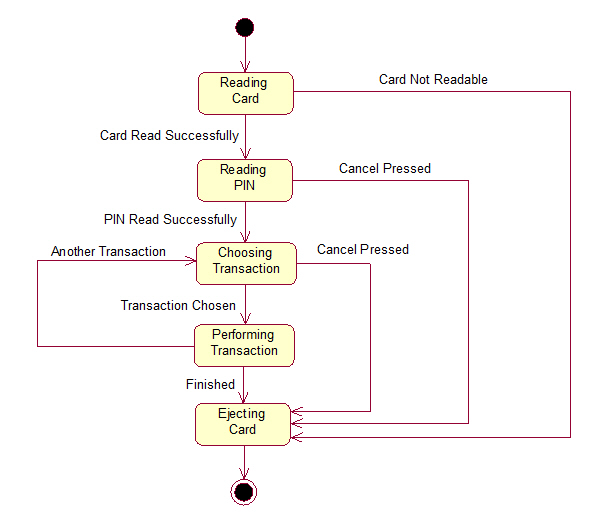


Figure 3: State-Chart Diagram for ATM machine

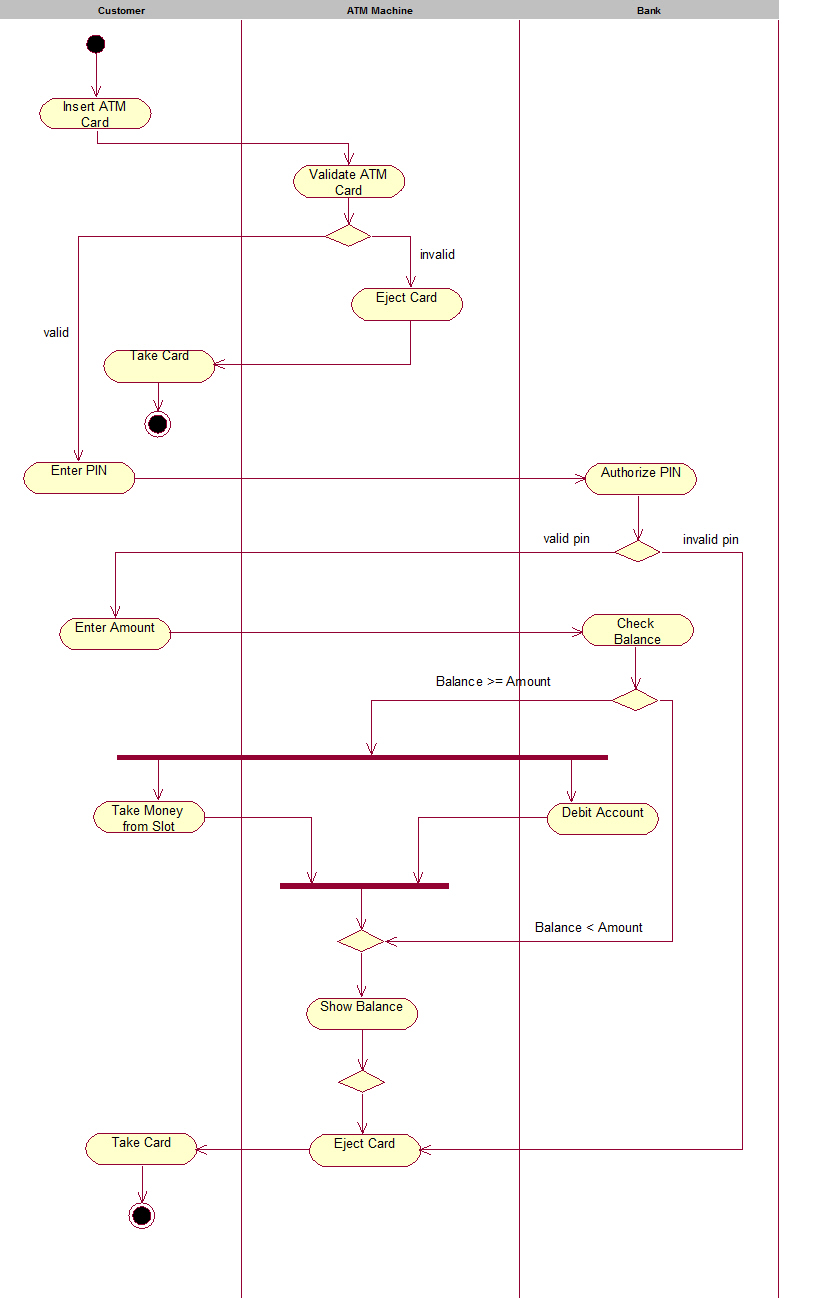
1. **Activity Diagram**

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 Checkings 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)

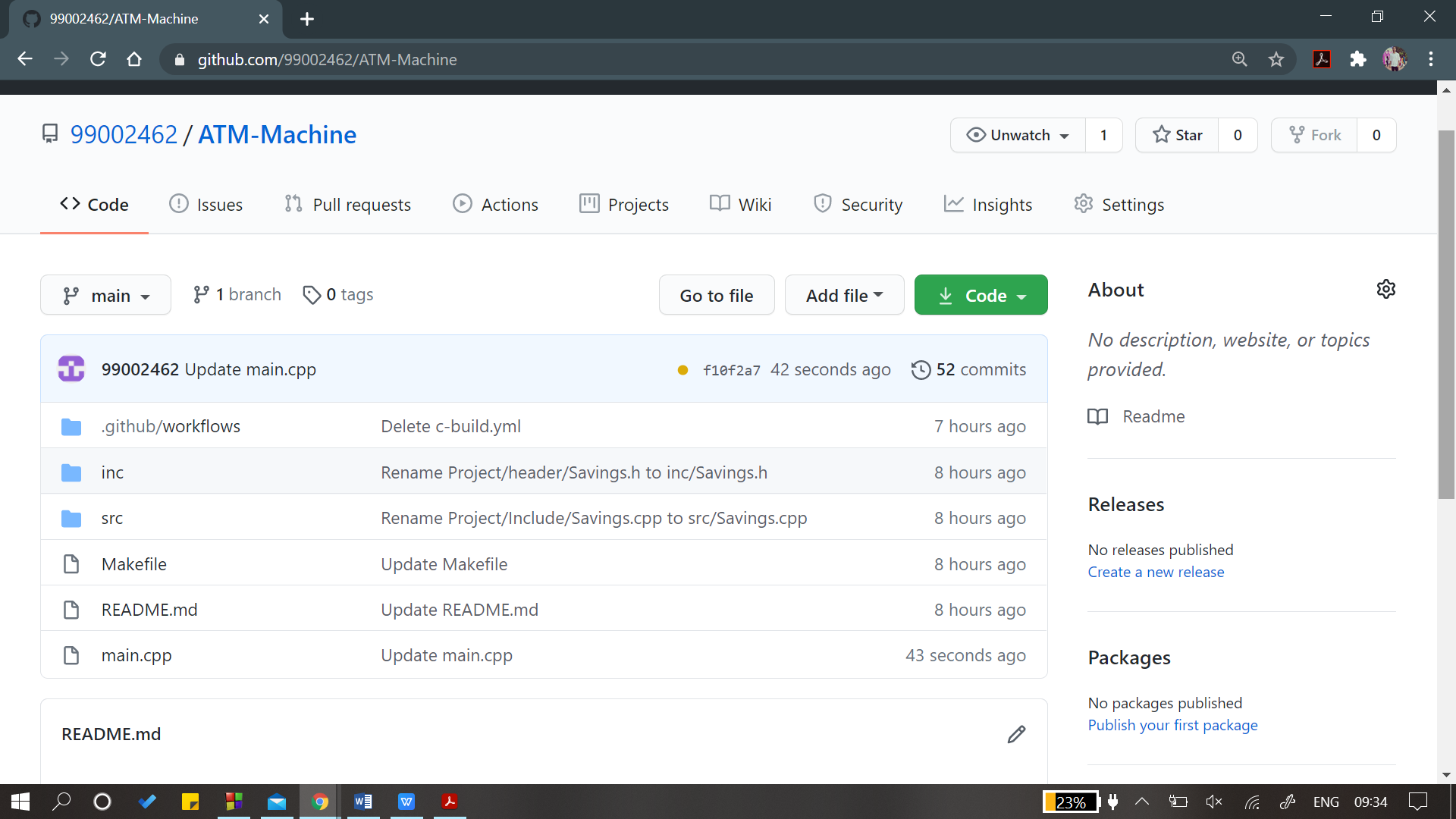


Figure 5: Repository View

### 

### 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)

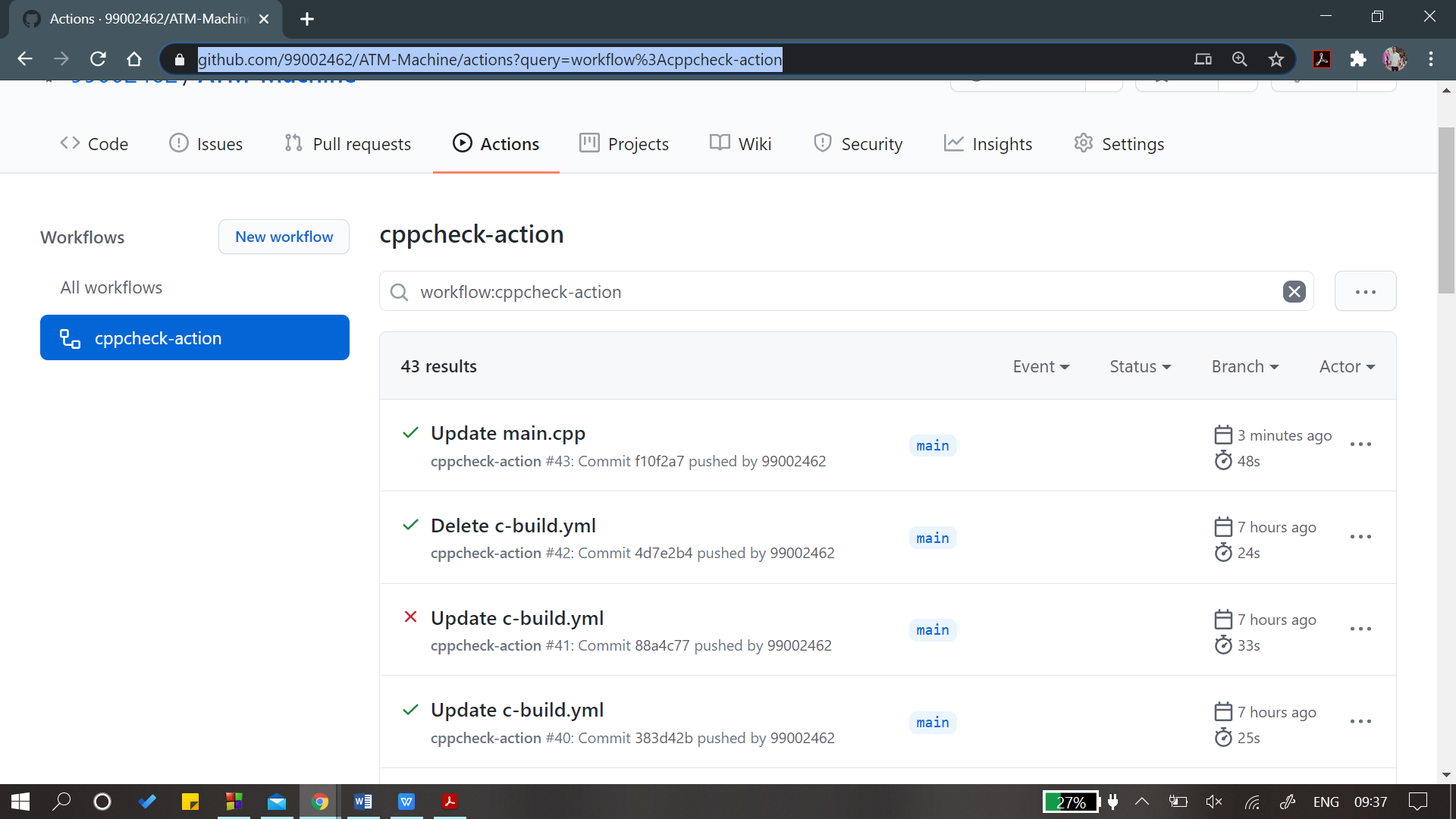


Figure 7: Cppcheck

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

Figure 8: Code Quality

# References

1. <https://en.wikipedia.org/>
2. <https://draw.io/>
3. <https://github.com/stepin654321/MiniProject_Template>
4. <https://www.startertutorials.com/uml/>