Learning Report –

SDLC and Testing Plan

**SMART TROLLEY-HUMAN GUIDED**

**INTELLIGENT CART SYSTEM**

Submitted By-

Abhay Kumar Dhar

P.S No-99002515

**CONTENTS:**

[FUTURE SCOPE: 4](#_Toc53000794)

[**HIGH LEVEL REQUIREMENTS**: 6](#_Toc53000795)

[FUNCTIONAL REQUIREMENT: 8](#_Toc53000796)

[ **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. 8](#_Toc53000797)

[NON FUNCTIONAL REQUIREMENT: 8](#_Toc53000798)

[Cart – Backend sync validation 15](#_Toc53000799)

[Single Product Remove from Cart 15](#_Toc53000800)

[Multiple Products Remove from Cart 15](#_Toc53000801)

[Multiple Products Remove from Cart 15](#_Toc53000802)

[Edit Quantity Cart boundary value 15](#_Toc53000803)

[Guest Check-out 16](#_Toc53000804)

[Registered 16](#_Toc53000805)

[Check Out 16](#_Toc53000806)

[Check-Out Mandatory 16](#_Toc53000807)

[Field Validation. 16](#_Toc53000808)

[**Activity 2** –CI Workflow for C Programming 20](#_Toc53000809)

[Introduction 20](#_Toc53000810)

[Requirements 20](#_Toc53000811)

[High level requirements 20](#_Toc53000812)

[Low level requirements 20](#_Toc53000813)

[Requirement Mapping 21](#_Toc53000814)

[Test plan mapping 21](#_Toc53000815)

[UML Diagram 22](#_Toc53000816)

[Use Case Diagram 23](#_Toc53000817)

[Activity Diagram 24](#_Toc53000818)

[GitHub screenshots 25](#_Toc53000819)

***Activity 1.1***

**Introduction**

**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.

**Abhay Kumar Dhar**

# 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. |

## **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-designed 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. |

## FUNCTIONAL REQUIREMENT:

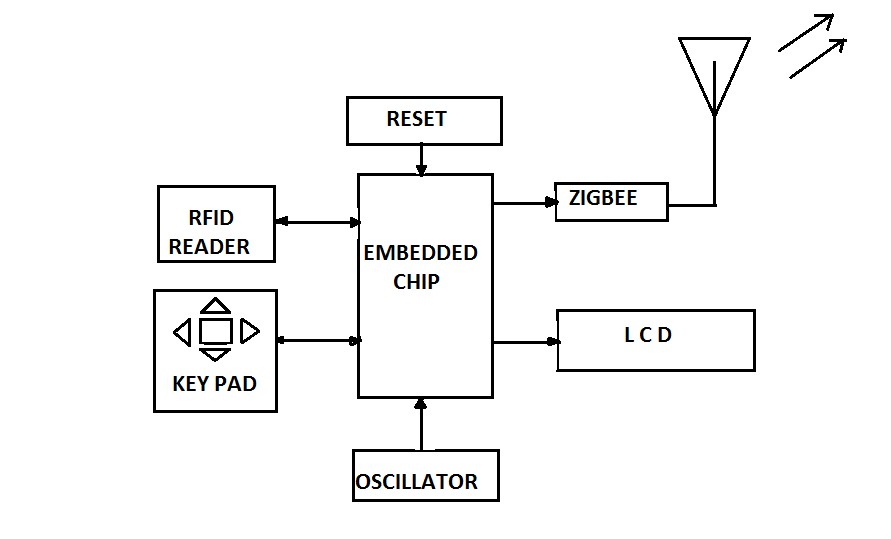
###  **Interface Requirement** -The system is capable to accept and transmit the raw data which may be in the form of digital that is numeric values.

* **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:

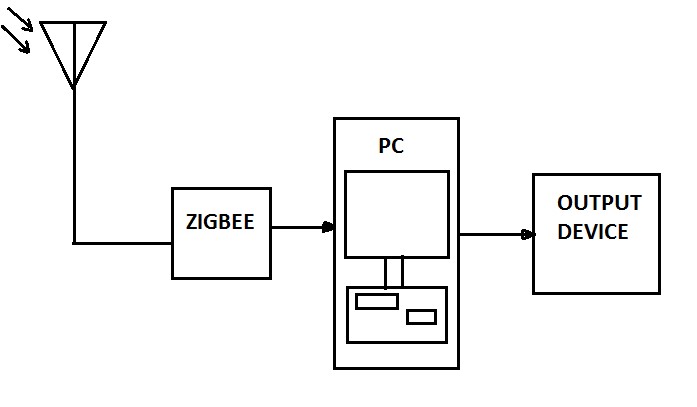
* **Maintainability:** Human resources is not required to maintain the components and collect the raw data from each of the components.
* **Reusability:** The components are compatible for changing environment and supports upgradeability.
* **Availability:** The system is functional throughout and data 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.***

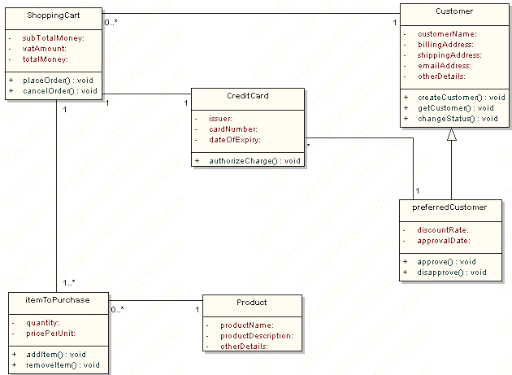


***Billing section.***

UML Diagrams:

Low Level:

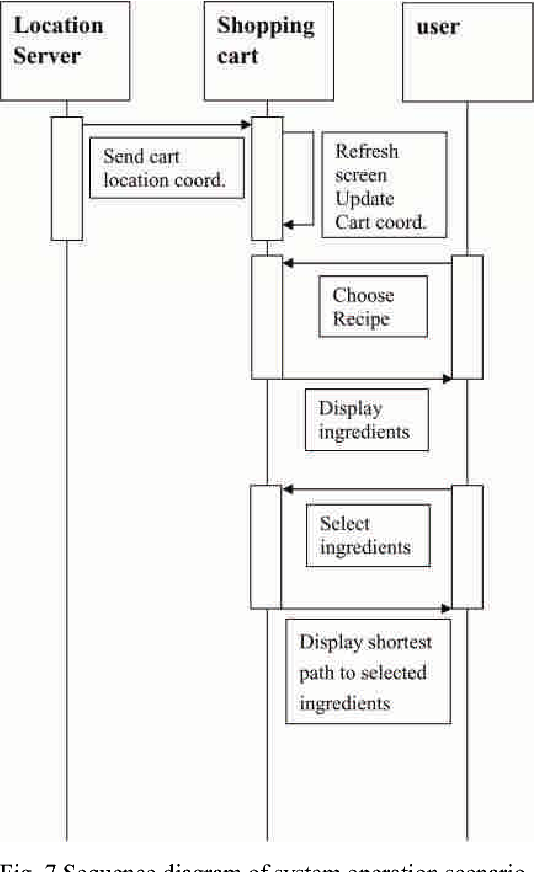
**Class Diagram:**



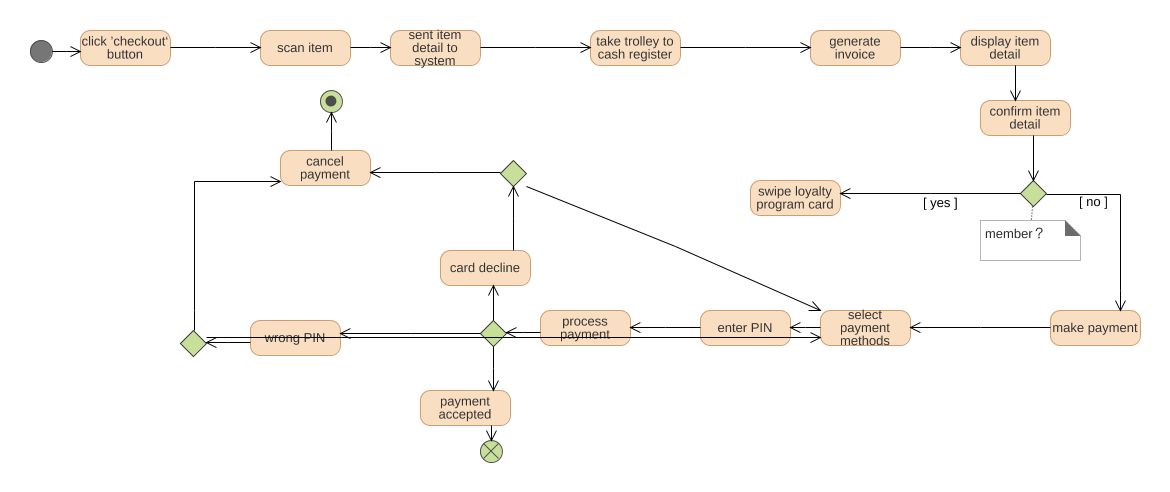
**Use case Diagram:**

****

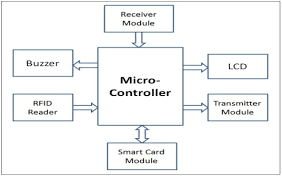
**High level:**

******

**Behavioral Diagram:**

******

**Functional Diagram:**

******

**High level 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. |

**Costing:**

|  |  |
| --- | --- |
| RFID reader | Rs.8900 |
| LCD Display | Rs.139 |
| ESP8266 | Rs.399 |
| Servo Motor | Rs.6000 |
| RFID Tags | Rs.249 |

**Aging:**

|  |  |
| --- | --- |
| Time | Gradation |
| PAST | Usually carry bags made of plastic or cloth, for heavy shopping metal basket |
| PRESENT | Shopping cart and handy basket is used |
| FUTURE | Automated shopping with smart system to do billing automatically |

**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.

**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 2** –CI Workflow for C Programming

Calculator

# Introduction

A calculator is a machine which allows people to do [math](https://simple.wikipedia.org/wiki/Mathematics) operations more easily. For example, most calculators will [add](https://simple.wikipedia.org/wiki/Addition), [subtract](https://simple.wikipedia.org/wiki/Subtraction), [multiply](https://simple.wikipedia.org/wiki/Multiplication), and [divide](https://simple.wikipedia.org/wiki/Division_(mathematics)). Some also do [square roots](https://simple.wikipedia.org/wiki/Square_root), and more complex calculators can help with [calculus](https://simple.wikipedia.org/wiki/Calculus) and draw function graphs. Calculators are found everywhere. A [smartphone](https://simple.wikipedia.org/wiki/Smartphone) or other [computer](https://simple.wikipedia.org/wiki/Computer) can also act as a calculator.

Some calculators, like the [abacus](https://simple.wikipedia.org/wiki/Abacus), will work without [batteries](https://simple.wikipedia.org/wiki/Battery). Others, like the electronic calculator, require batteries. There are two types of electronic calculators: simple calculators, which can only [add](https://simple.wikipedia.org/wiki/Addition), [subtract](https://simple.wikipedia.org/wiki/Subtraction), [multiply](https://simple.wikipedia.org/wiki/Multiplication) and [divide](https://simple.wikipedia.org/wiki/Division_(mathematics)), and sometimes take [square roots](https://simple.wikipedia.org/wiki/Square_roots); and scientific calculators, which can do many other things, such as calculate [factorials](https://simple.wikipedia.org/wiki/Factorial) and [trigonometry](https://simple.wikipedia.org/wiki/Trigonometry) functions.

# Requirements

## High level requirements

1. Performance

The performance of the calculator should be high.

1. Speed

The speed of operations performed should be fast.

1. It should perform all the arithmetic operations.
2. It should find the area of square.
3. It should perform conversion operations.
4. It should find the factorial of number.
5. It should check for prime number.

## Low level requirements

1. It should take operands as input and give the result of addition, subtraction, multiplication and division of the operands accordingly.
2. It should take two operands as input and give the area of square.
3. It should take a operand as input and make conversion from kilometer to meter, centimeter and millimeter.
4. It should take the operand as input and give its factorial as output.
5. It should take a operand as input check if it is prime number.

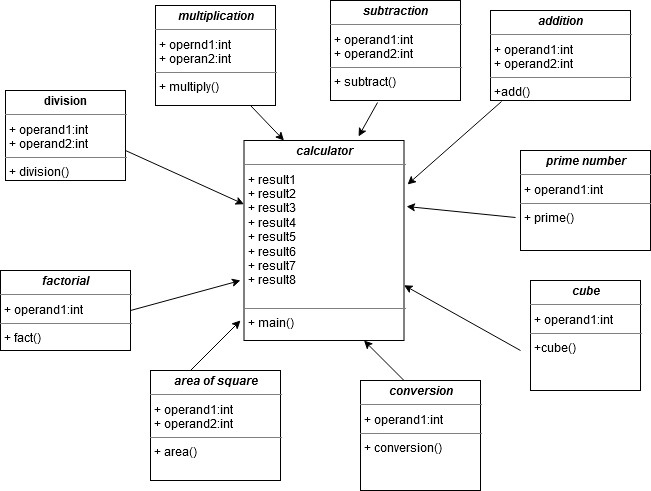
# Requirement Mapping

|  |  |
| --- | --- |
| ID | Description |
| H\_01 | Performance  The performance of the calculator should be high. |
| H\_02 | Speed  The speed of operations performed should be fast. |
| H-03\_L\_01 | It should perform all the arithmetic operations.  It should take operands as input and give the result of addition, subtraction, multiplication and division of the operands accordingly. |
| H-04\_L\_02 | It should find the area of square  It should take two operands as input and give the area of square. |
| H-05\_L\_03 | It should perform conversion operations.  It should take a operand as input and make conversion from kilometer to meter, centimeter and millimeter. |
| H-06\_L\_04 | It should find the factorial of number.  It should take the operand as input and give its factorial as output. |
| H-07\_L\_05 | It should check for prime number.  It should take a operand as input check if it is prime number. |

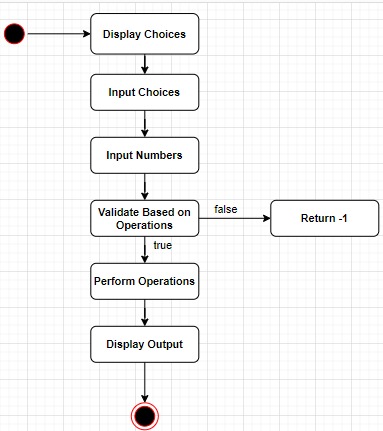
# Test plan mapping

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Description | Precondition | Expected input | Expected output | Actual Output |
| H\_01 | Performance | Should be 90% and above | - | High performance | - |
| H\_02 | speed | <20ms | - | Speed<20ms | <20ms |
| H\_03 | addition | Two operands as input | 2 and 3 | 5 | 5 |
| H\_04 | subtraction | Two operands as input | 3 and 2 | 1 | 1 |
| H\_05 | multiplication | Two operands as input | 2\*3 | 6 | 6 |
| H\_06 | division | Two operands as input | 4/2 | 2 | 2 |
| H\_07 | factorial | One operand | 3 | 6 | 6 |
| H\_08 | Conversion from kilometer to meter | One operand | 100km | 100000m | 100000m |
| H\_09 | Area of square | Two operands as input | 2 and 2 | 4 | 4 |

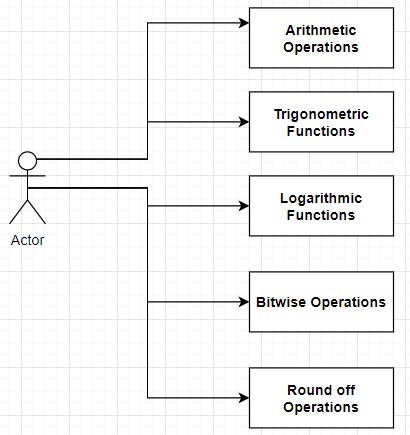
# UML Diagram



# Use Case Diagram

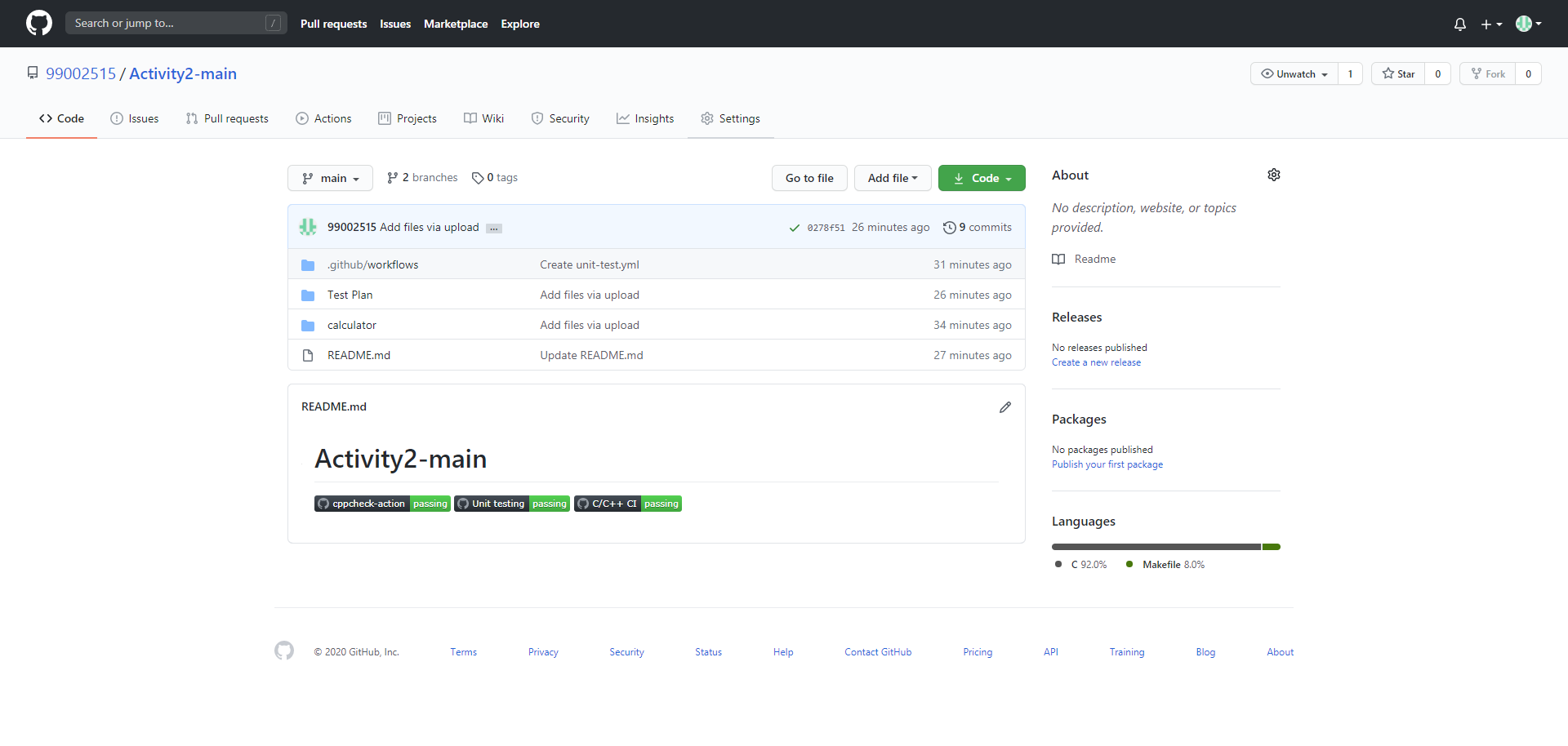


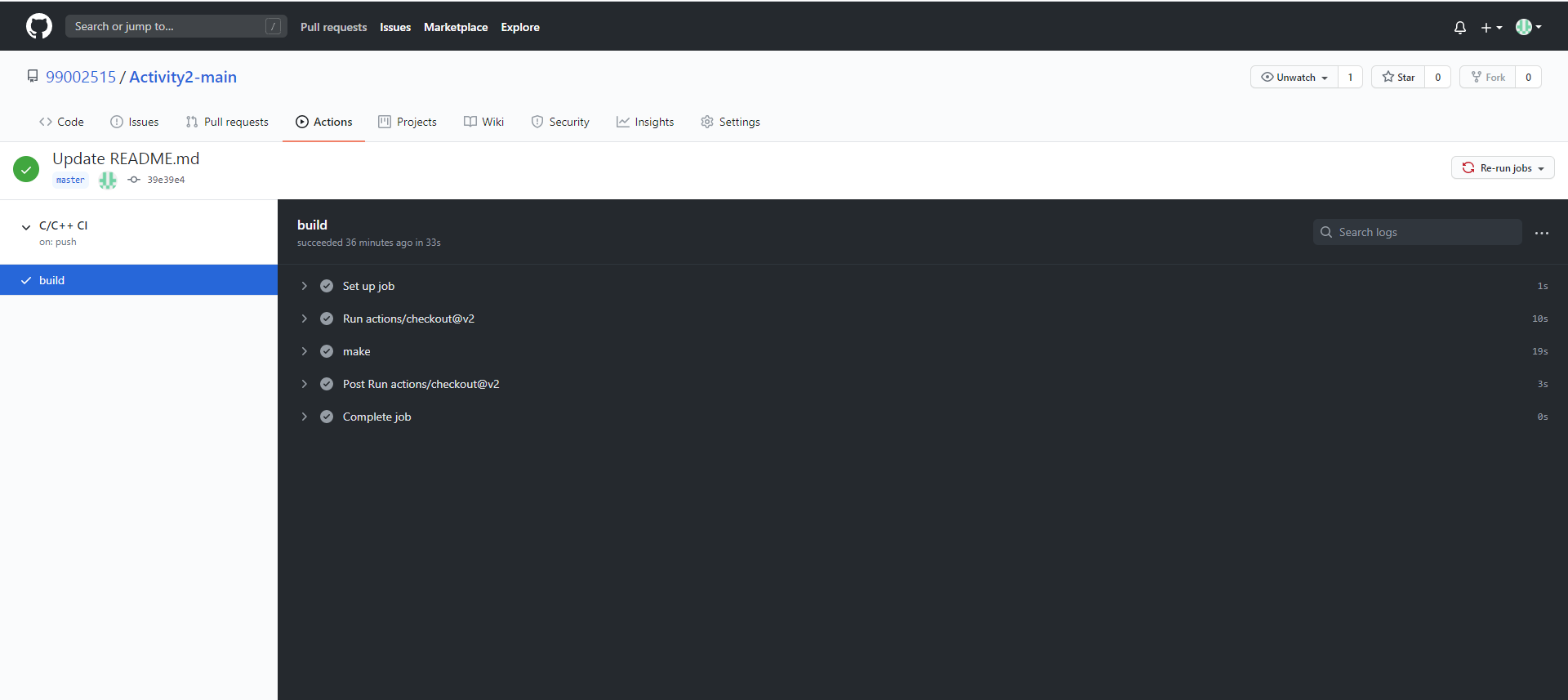
# Activity Diagram



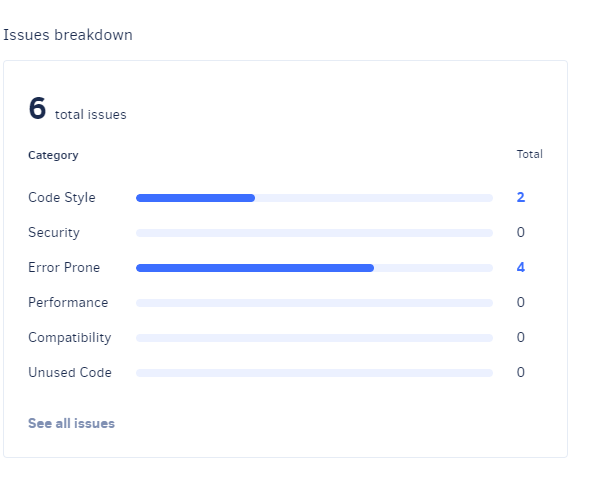
## GitHub screenshots

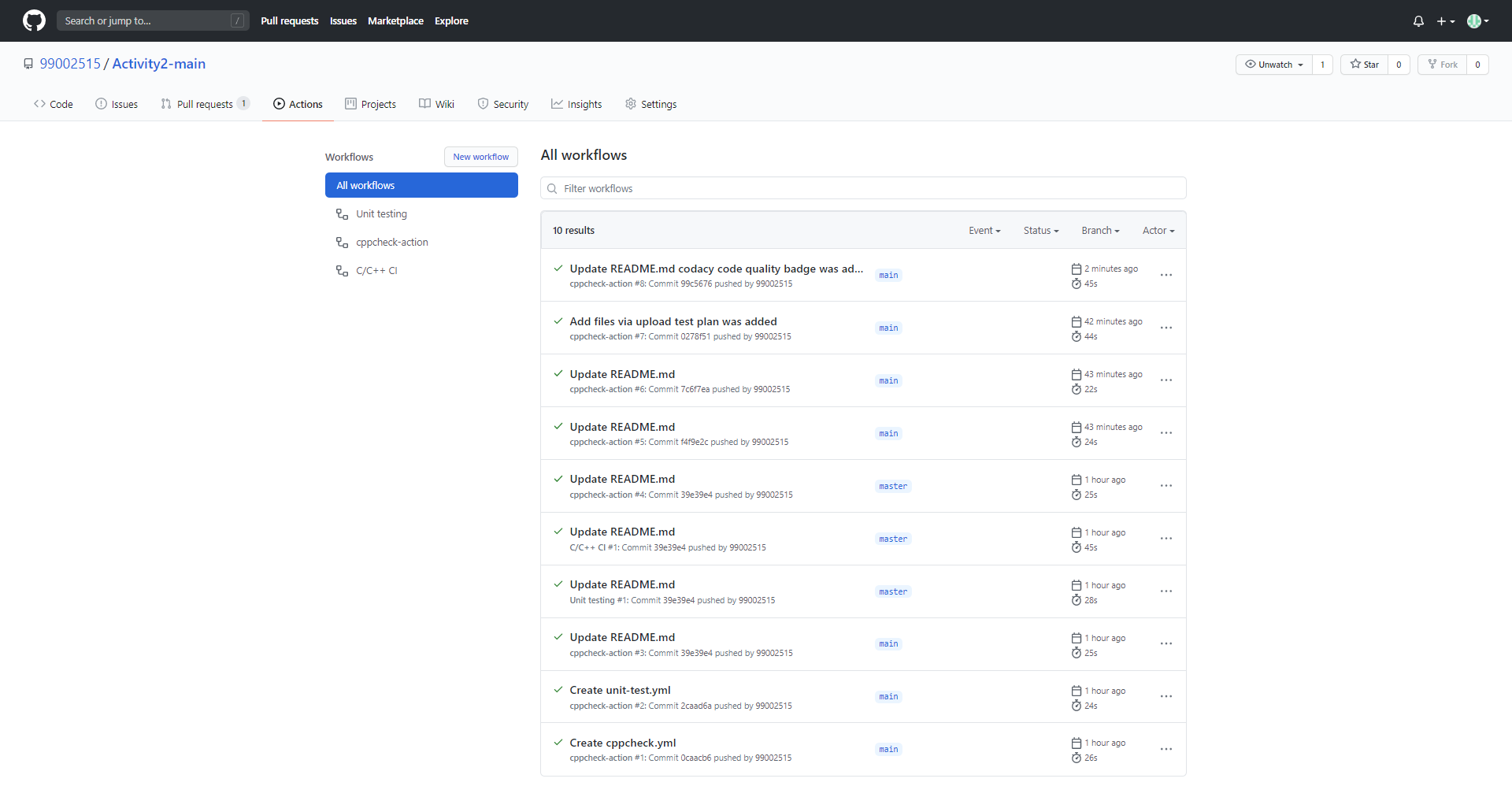
Badges



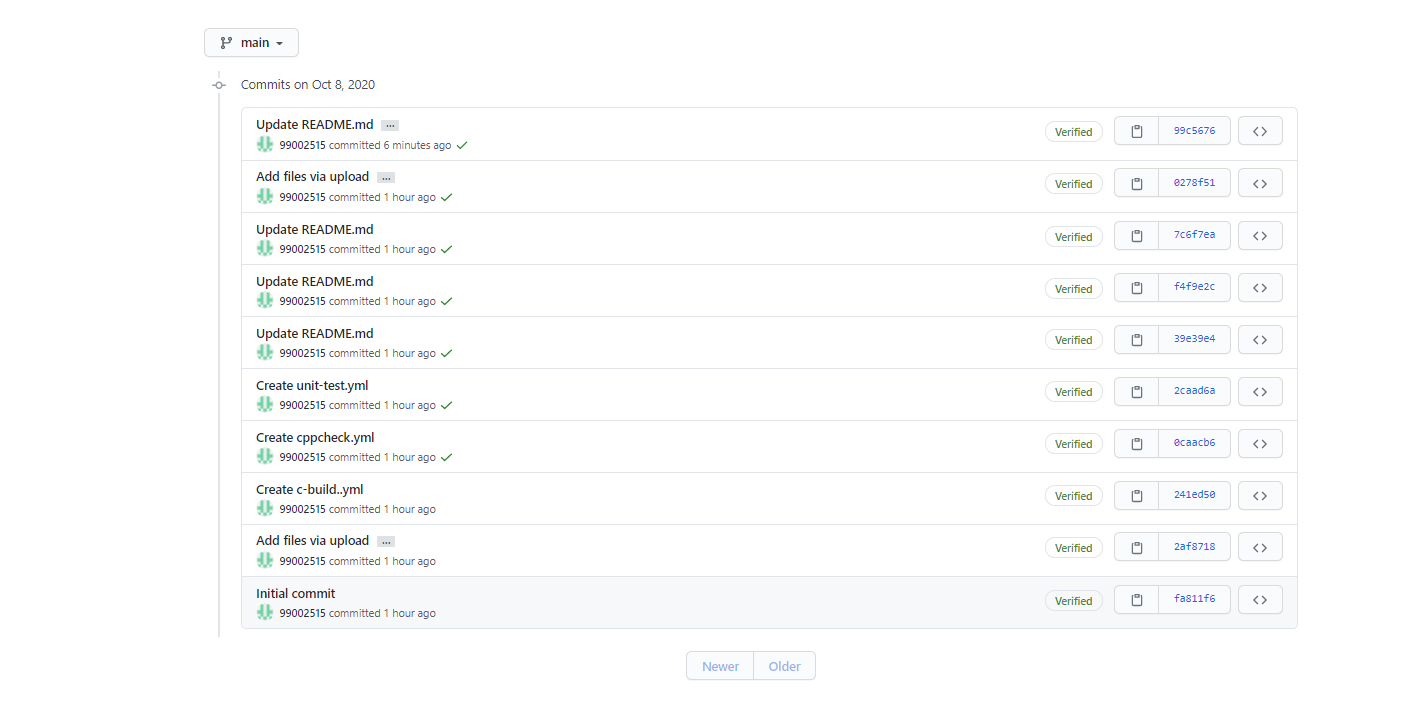


Codacy analysis:





GitHub link - <https://github.com/99002515/Activity2-main>



Github Commit Link- <https://github.com/99002515/Activity2-main/commits/main>