Learning Report –

SDLC and Testing

A Report By-

Name- Saubhagya Ashish
Ps No- 99002607

Table of Content

ACTIVITY -1

Product Selected - Try On Augmented Reality Watch

Ageing -

- Augmented reality technology was invented in 1968, with Ivan Sutherland's development of the first head-mounted display system. However, the term 'augmented reality' wasn't coined until 1990 by Boeing researcher Tim Caudell.

- A view of the physical real-world environment with superimposed Computer-generated images, thus changing the perception of reality, is the AR.

- According to Apple CEO Tim Cooke, Augmented Reality is the core technology and will be big technological step forward, which is similar to the release of smartphones.

Costing of Products-

- Earlier due to shortage of skills a basic augmented application cost around $100 - $200.
- After invention of software like Unity and Spark Augmented and Virtual Reality become quite easy to make.
- In present augmented reality is used almost every platform like e commerce, medical, and etc.
- It is one of the largest employment sector in the world right now.

SWOT Analysis-

Strength-
- Application creates the virtual objects.
- Virtual object well defines the original object.

Weakness-
- The application is virtual so can't be filled or touched.
- The application only runs android platform

Opportunities-
- It gives customer a detail look of the product
- With new updates the product can generate a lot of revenue.

Threats-
- Security concerns are one of major issues.

Requirements -

High Level Requirements -

- Android Smartphone
- Minimum Android Version- 8.0
- Minimum Storage Space of 45 MB
- In build Camera
- Image Target
- Virtual Object Formation

Low Level Requirements -

- Creating Core Functionality -
  - Create 3 different Watch Models
  - Occlusion Of hand

- Creating User Interface -
  - Create UI slide in Frame
  - Create Color switch buttons for watches
  - Create a Exit Button

DESIGN

High Level Design -

Structural Diagram -

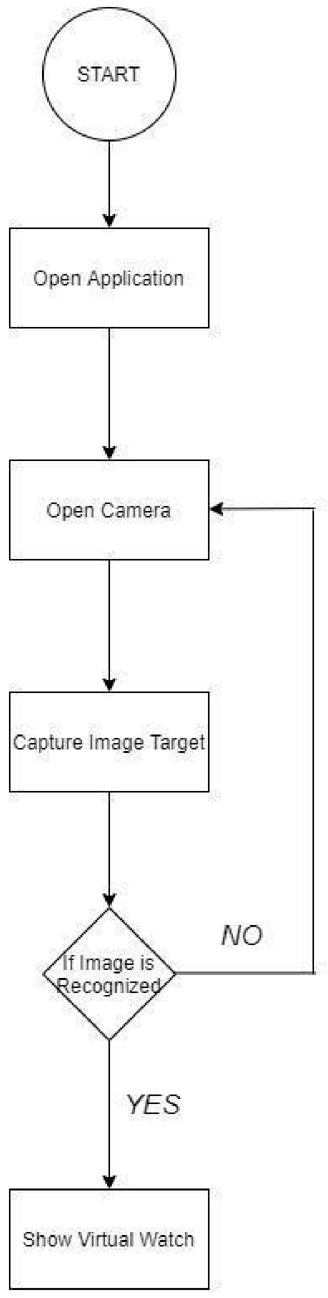
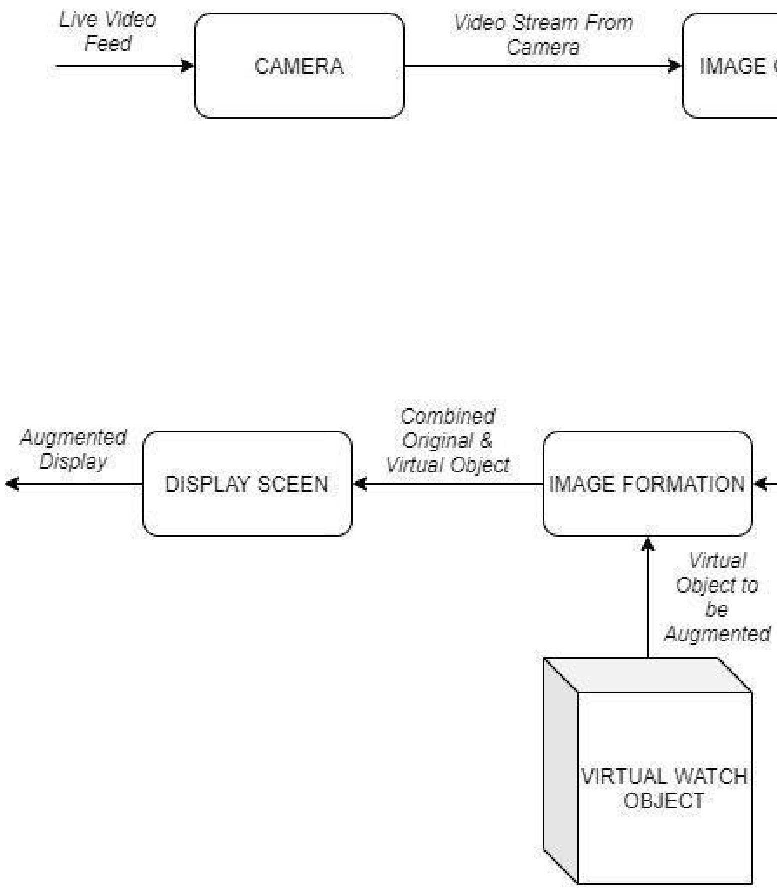


Fig-1 Flow Chart Diagram

Behavioral Diagram

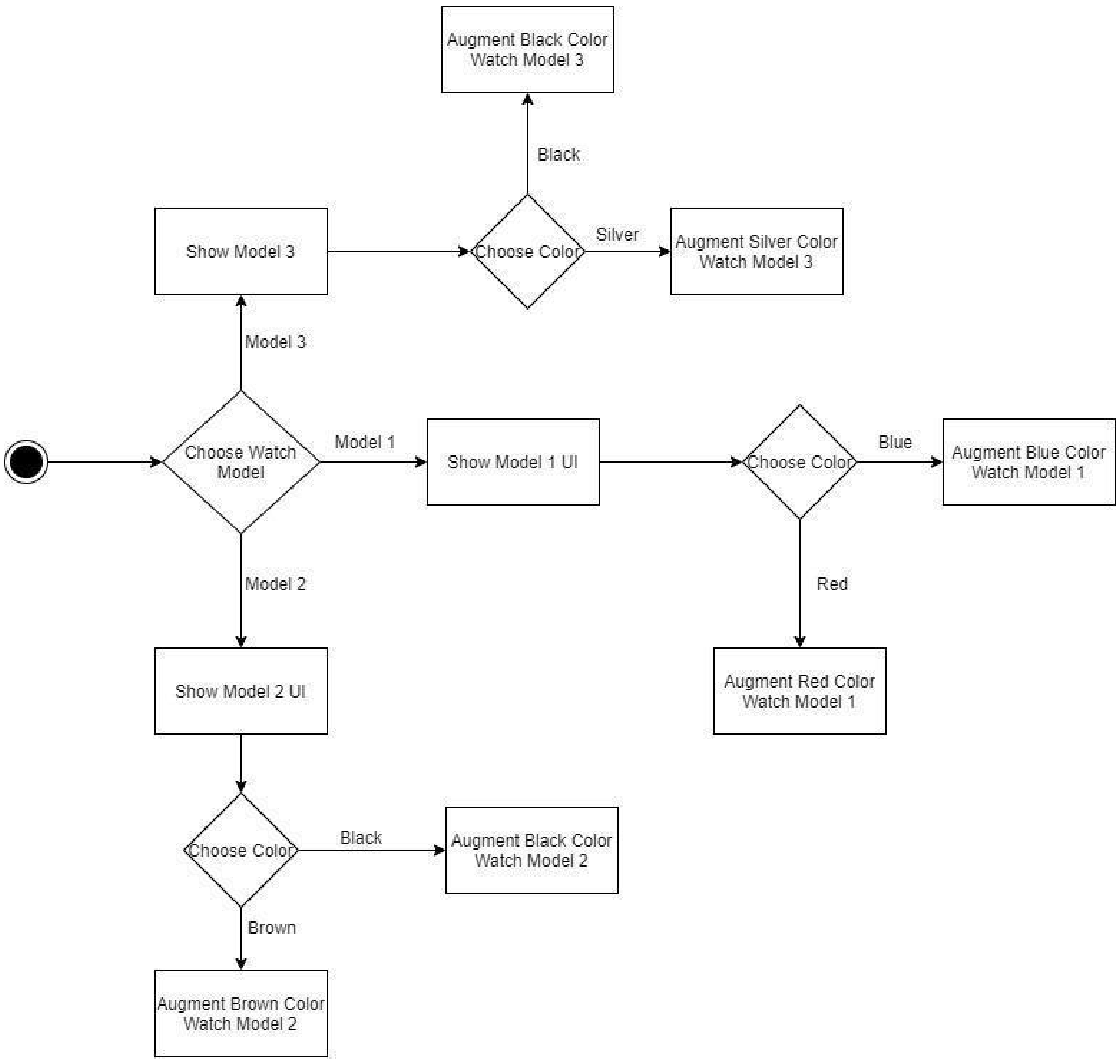
- State Chart Diagram



State Chart Diagram

Low Level Design-

Structural Diagram -



Augment Black Color Watch Model 3
Black
Show Model 3
Choose Color
Silver
Augment Silver Color Watch Model 3
Model 3
Choose Watch Model
Model 1
Show Model 1 UI
Choose Color
Blue
Augment Blue Color Watch Model 1
Model 2
Red
Show Model 2 UI
Augment Red Color Watch Model 1
Choose Color
Black
Augment Black Color Watch Model 2
Brown
Augment Brown Color Watch Model 2

Fig 3: Flow Chart Diagram

Behavioral Diagram -

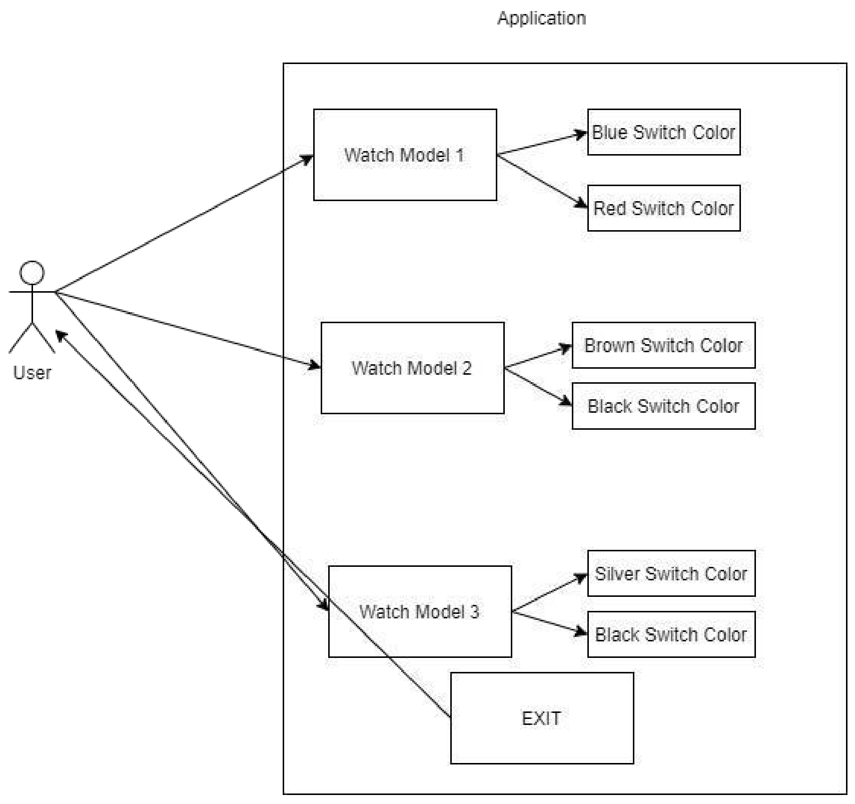


Fig 4: Use Case Diagram

Testing -

TEST ID	DESCRIPTION	EXPECTED INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
T001	Animation of Watch window 1	Tap on Window 1 UI	Watch window 1 slides into the frame	Watch window 1 slides into the frame
T002	Animation of Watch window 2	Tap on Window 2 UI	Watch window 2 slides into the frame	Watch window 2 slides into the frame
T003	Animation of Watch window 2	Tap on Window 3 UI	Watch window 3 slides into the frame	Watch window 3 slides into the frame
T004	Exit Button of Watch window 1	Tap on Exit Button of Watch window 1	Watch window 1 slides back from the frame	Watch window 1 slides back from the frame
T005	Exit Button of Watch window 2	Tap on Exit Button of Watch window 2	Watch window 2 slides back from the frame	Watch window 2 slides back from the frame
T006	Exit Button of Watch window 3	Tap on Exit Button of Watch window 3	Watch window 3 slides back from the frame	Watch window 3 slides back from the frame
T007	Color Switch Buttons of Watch window 1	Tap on the Red color switch buttons	Color of Watch Changes to Red	Color of the Watch Changes to Red
T008	Color Switch Buttons of Watch window 1	Tap on the Blue color switch buttons	Color of Watch Changes to Blue	Color of Watch Changes to Blue
T009	Color Switch Buttons of Watch window 2	Tap on the Black color switch buttons	Color of Watch Changes to Black	Color of Watch Changes to Black
T010	Color Switch Buttons of Watch window 2	Tap on the Brown color switch buttons	Color of Watch Changes to Brown	Color of Watch Changes to Brown
T011	Color Switch Buttons of Watch window 3	Tap on the Black color switch buttons	Color of Watch Changes to Black	Color of Watch Changes to Black
T012	Color Switch Buttons of Watch window 3	Tap on the Silver color switch buttons	Color of Watch Changes to Silver	Color of Watch Changes to Silver

ACTIVITY -2

CALCULATOR

Introduction

A calculator is a machine which allows people to do math operations more easily. For example, most calculators will add, subtract, multiply, and divide. Some also do square roots, and more complex calculators can help with calculus and draw function graphs. Calculators are found everywhere. A smartphone or other computer can also act as a calculator.
Some calculators, like the abacus, will work without batteries. Others, like the electronic calculator, require batteries. There are two types of electronic calculators: simple calculators, which can only add, subtract, multiply and divide, and sometimes take square roots; and scientific calculators, which can do many other things, such as calculate factorials and trigonometry functions.

Requirements

High Level Requirement

- Performance - The performance of the calculator should be high.
- Speed -The speed of operations performed should be fast.
- It should perform all the arithmetic operations.
- It should find the area of square.
- It should perform conversion operations.
- It should find the factorial of number.
- It should check for prime number.

Low level requirements

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

1. 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 kilometerto meter	One operand	100km	100000m	100000m
H_09	Area of square	Two operands as input	2 and 2	4	4

UML Diagram

1. Class Diagram

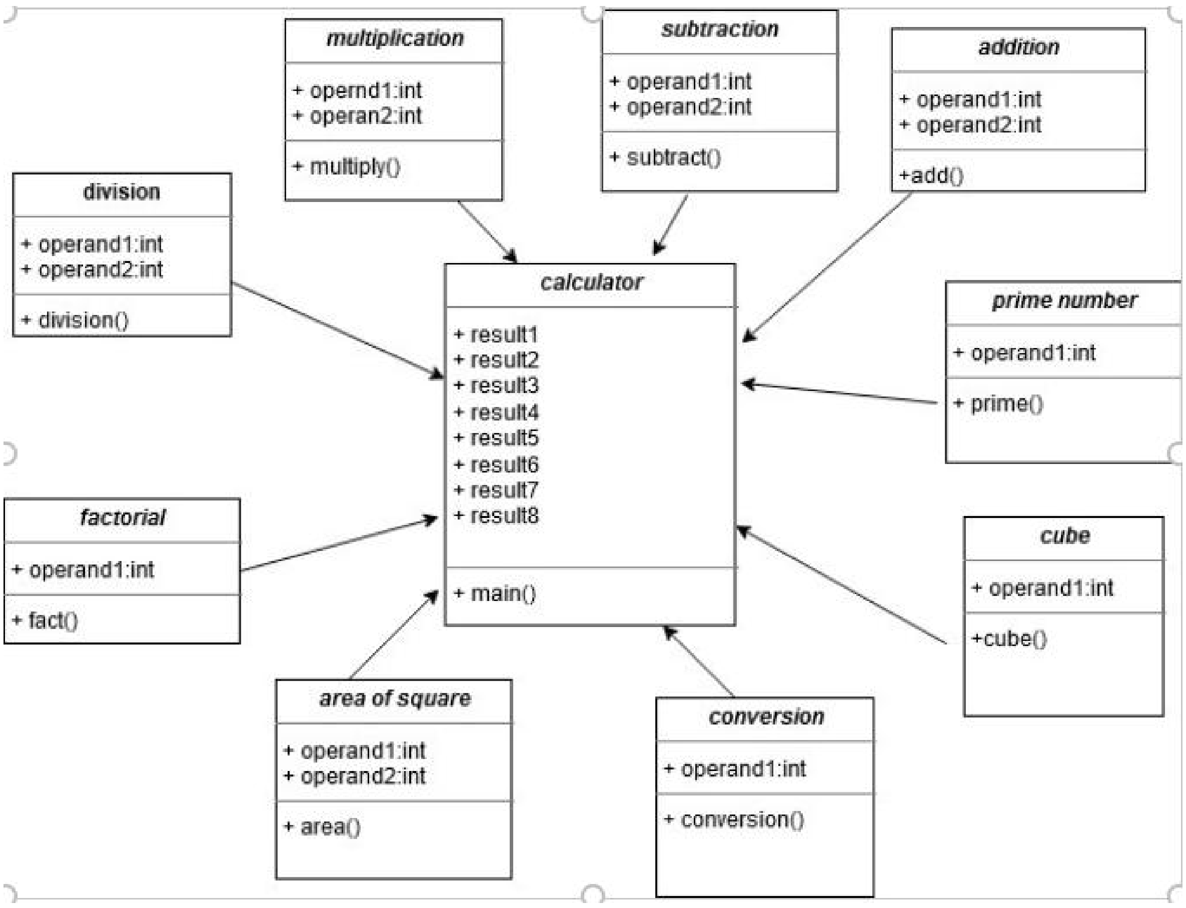


Fig. 5: Class Diagram

2. Use Case Diagram

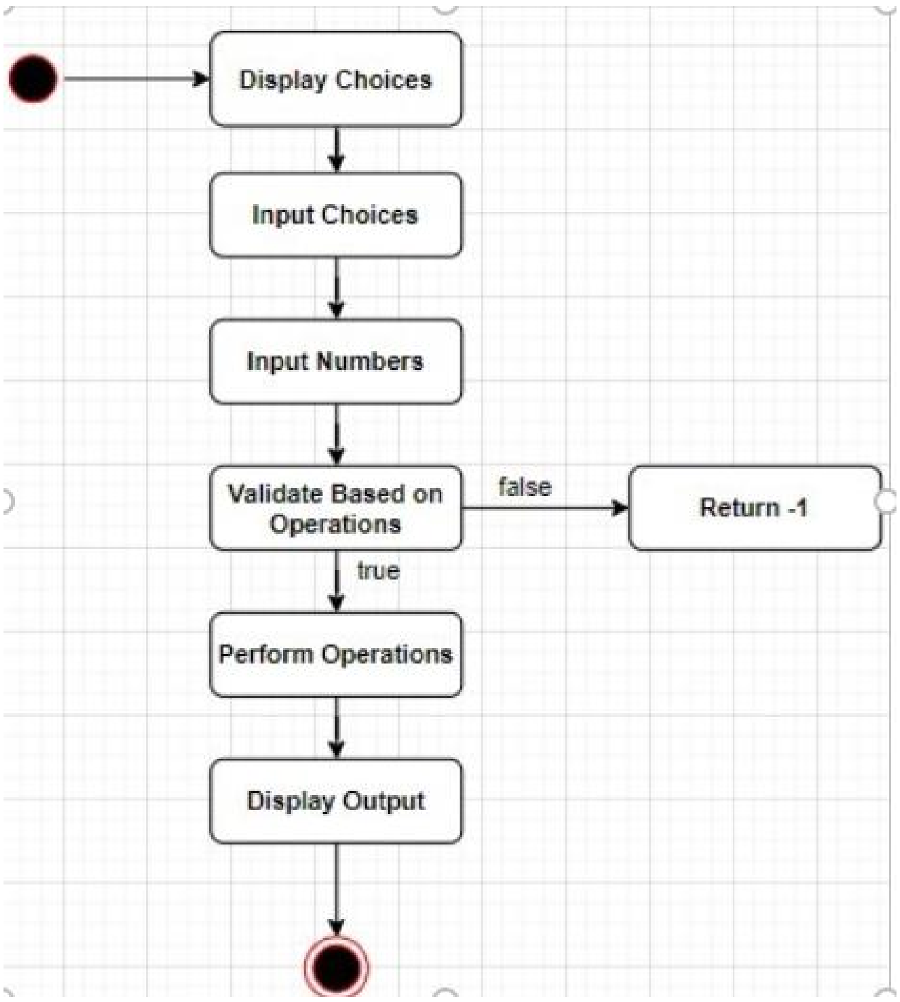


Fig 6: Use Case Diagram

3. Activity Diagram

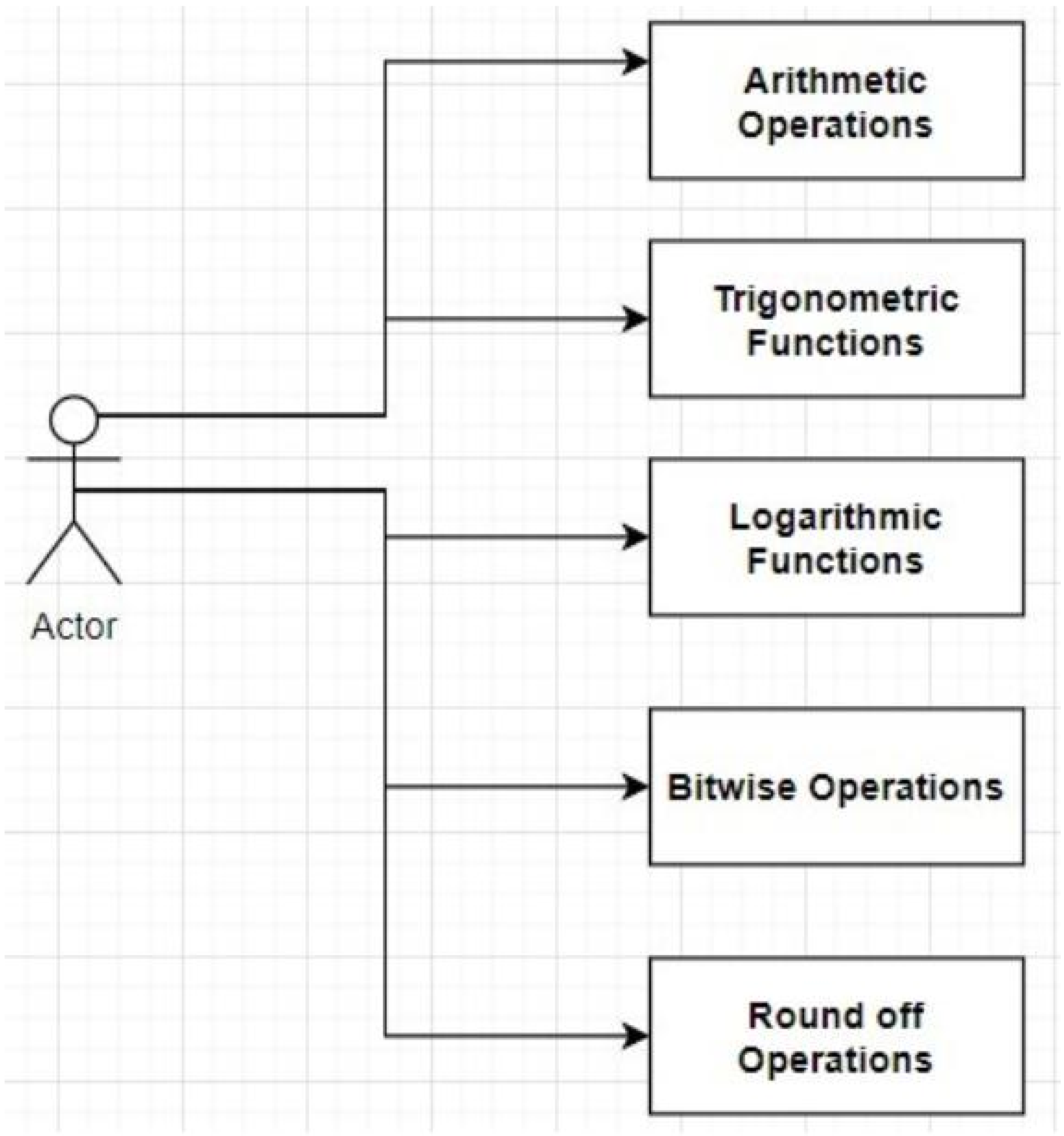
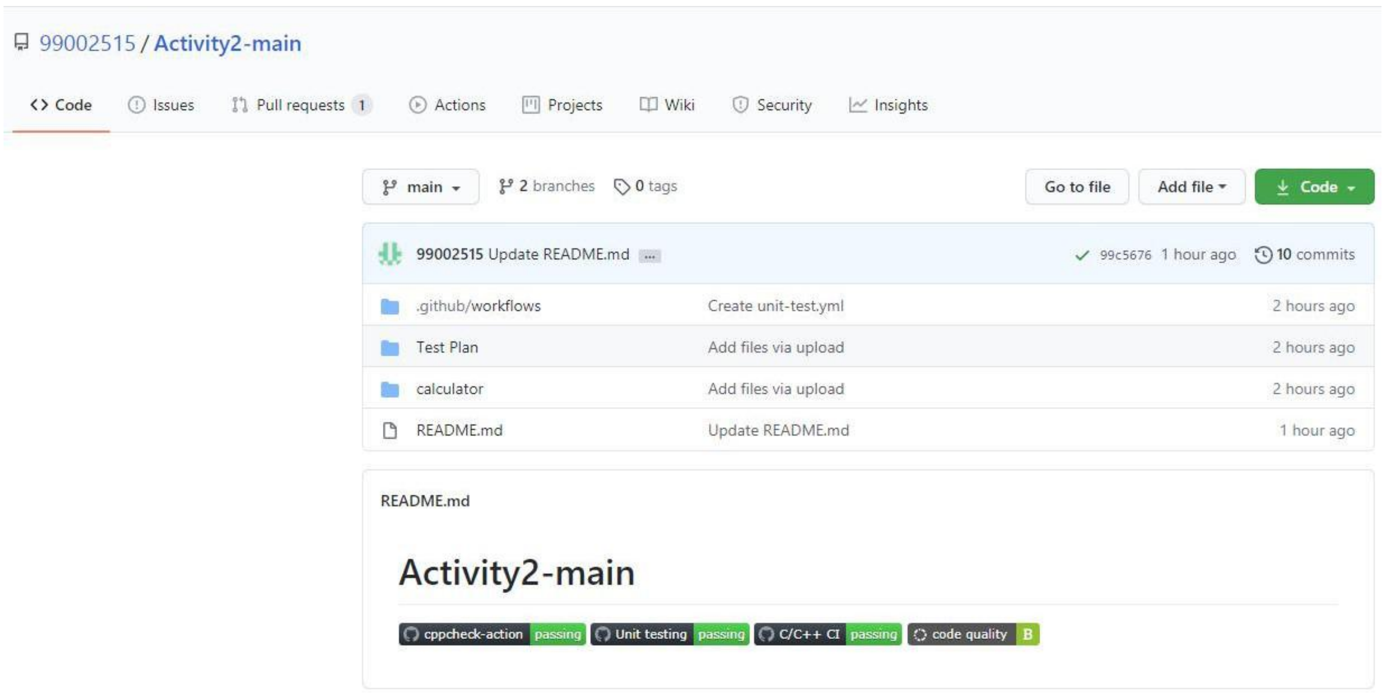


Fig 7: Activity Diagram

GitHub screenshots

Badges


99002515 / Activity2-main

99002515 Update README.md ... ✓ 99c5676 1 hour ago 10 commits

.github/workflows	Create unit-test.yml	2 hours ago
Test Plan	Add files via upload	2 hours ago
calculator	Add files via upload	2 hours ago
README.md	Update README.md	1 hour ago

README.md

Activity2-main

cppcheck-action passing | Unit testing passing | C/C++ CI passing | code quality B

GitHub link - https://github.com/99002607/Applied_SDLC

ACTIVITY – 3 (Agile Implementation)

Theme: Calculation

Epic 1: Simple math operations.

User Story1:

- As an accountant.
- I want to add 2 large numbers.
- I want to see the accurate result instantly.

Test Case:

- Given 2 numbers 9055 and 53245.
- When I add them.
- Result should be 62300 within no time.

User Story 2:

- As a Student.
- I need to perform mathematical operations.
- I need accurate and precise results.

Test Case 1:

- Given any number.
- I need explicit option to perform all basic operations.
- And the result should be accurate.

Epic 2: Ease of Operation

User Story 1:

- As a Student.
- I want to see all the calculations that I made from when I switched on the calculator.
- So that I can backtrack my calculations.

Test Case 1:

- Given I am in the middle of an operation.
- When I press ‘repeat’ key.
- Then the calculator must display all my previous calculations.

Test Case 2:

- Given that this is my first calculation.
- When I press 'repeat’ key.
- The calculator must display 0.

Epic 3: Calculating Factorial

User Stories:

- As an engineering student
- I want to do factorial of a number.

Test Cases:

- Given Inputs for number
- Generate factorial of that.

Areas of Love and Challenge

Love

=> New way of implementing same project

Challenge

=> Thinking of different epic was bit difficult

Activity 4(MakeFile)

GitHub Link:

- https://github.com/99002607/Make_File

References

- https://www.programiz.com/c-programming/examples/calculator-switch-case
- https://www.tutorialspoint.com/git/index.htm
- https://www.codacy.com/products/how-to-do-code-review
- https://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/
- https://www.cs.swarthmore.edu/~newhall/unixhelp/howto_makefiles.html