./

Learning Report – GENESIS Week1 Final Submission

Course Code: <CODE>



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **Approved By** | **Remarks/Revision Details** |
| 1 | 18/09/20 | Milind Mohapatra |  |  |  |
| 2 | 19/09/20 | Milind Mohapatra |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

# 

Contents

[Activity and Tasks 4](#_Toc51407795)

[Activity 1– System/Software Development 4](#_Toc51407796)

[Activity 2 – Agile Aspects 10](#_Toc51407797)

[Activity 3– System/Software Development for Genesis week1 final submission 11](#_Toc51407798)

[Activity 2 – Agile Aspects 14](#_Toc51407799)

[Activity 3 15](#_Toc51407800)

[–CI Workflow for C Programming 15](#_Toc51407801)

# Activity and Tasks

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

* Description:

Automatic street lamp controller is a system that senses its surroundings and controls the street lamps accordingly. The system takes ambient light as input and toggles the street lamps as required. This helps in conserving energy and only turning ON the lamps whenever necessary.

* Requirements:
* High level requirements:

|  |  |
| --- | --- |
| **ID** | **Description** |
| HL\_01 | Lamp must turn ON when ambient light is below threshold |
| HL\_02 | Lamp must turn OFF when ambient light is above threshold |
| HL\_03 | System must function regardless of weather conditions |
| HL\_04 | Battery life must be long |

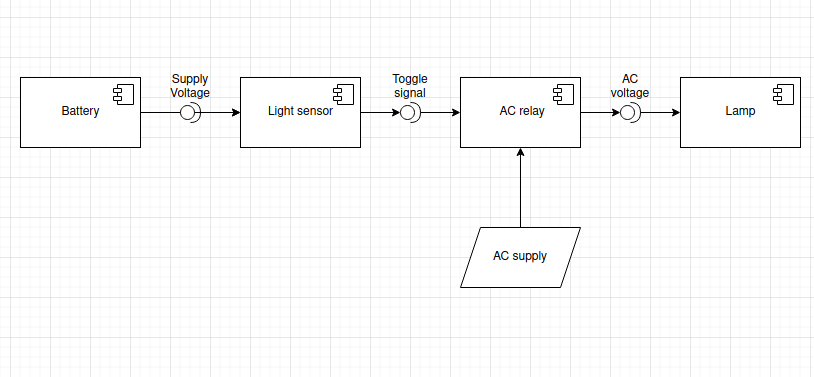
* Low level requirements:

|  |  |
| --- | --- |
| **ID** | **Description** |
| LL\_01 | Ambient light above threshold must result in HIGH signal from amplifier |
| LL\_02 | Below threshold ambient light must result in a LOW signal from amplifier |
| LL\_03\_1 | Lamp must turn ON during rain |
| LL\_03\_2 | Lamp must turn ON during fog |
| LL\_04 | Operating voltage must be low |

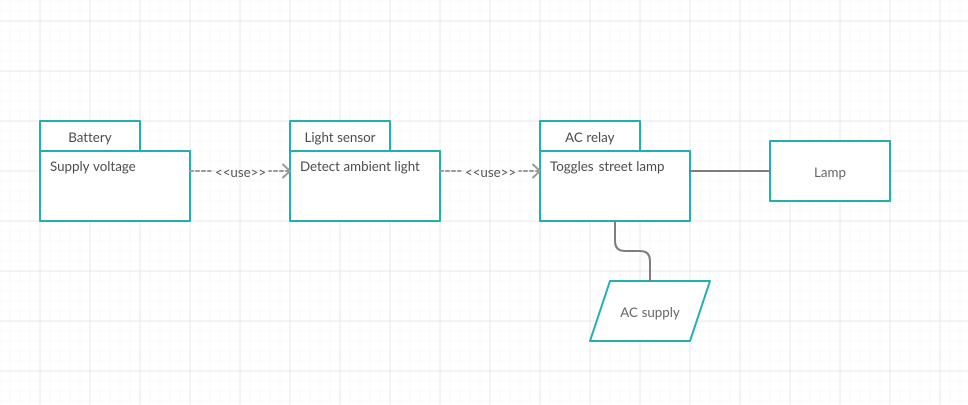
* Test plan:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Precondition** | **Expected I/P** | **Expected O/P** | **Actual O/P** |
| HL\_01 | Check lamp status when ambient light is LOW | System has power supply | Ambient light | Lamp turns ON |  |
| HL\_02 | Check lamp status when ambient light is HIGH | System has power supply | Ambient light | Lamp turns OFF |  |
| HL\_03\_1 | Lamp status on a rainy/foggy day | System has power | Ambient light | Lamp turns ON |  |
| HL\_03\_2 | Lamp status at night with many vehicles | System has power; night/cloudy | Ambient light | Lamp turns ON |  |
| LL\_01 | Check amplifier output when ambient light is LOW | Amplifier has power supply | Signal from LDR | Lamp turns ON |  |
| LL\_02 | Check amplifier output when ambient light is HIGH | Amplifier has power supply | Signal from LDR | Lamp turns OFF |  |
| LL\_03 | Operating voltage | System has power supply | Supply from battery | Power consumption is low |  |

* Design
  + High level UML design
    - Structural diagram:

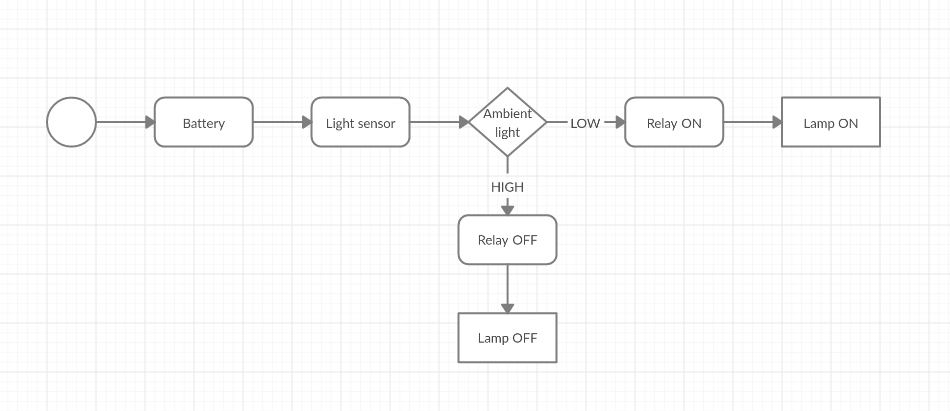


Component diagram

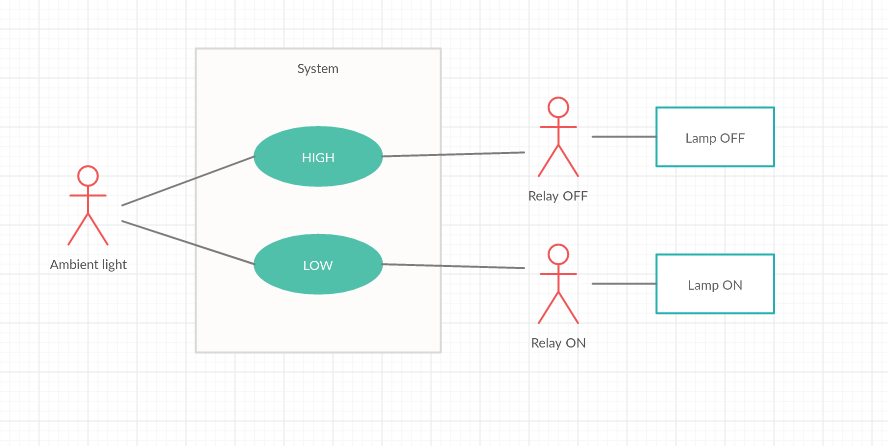


Package diagram

* + - Behavioral diagram:

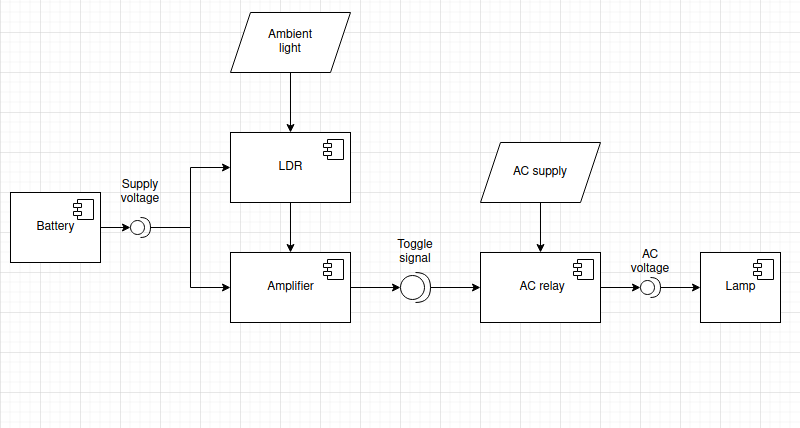


Activity diagram

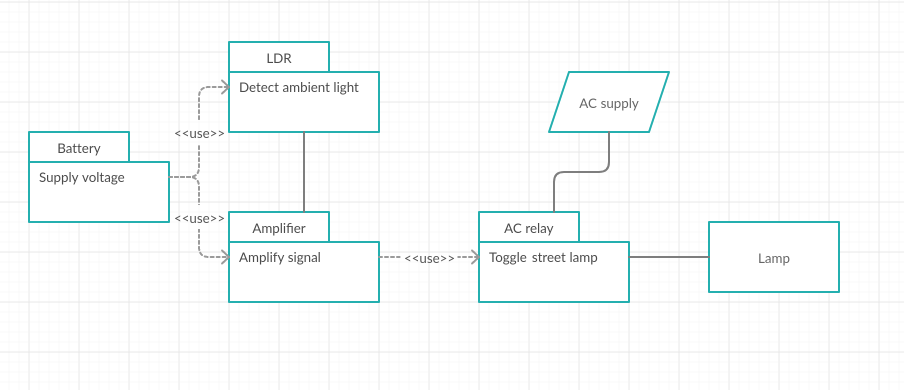


Use case diagram

* + Low level UML diagram
    - Structural diagram:

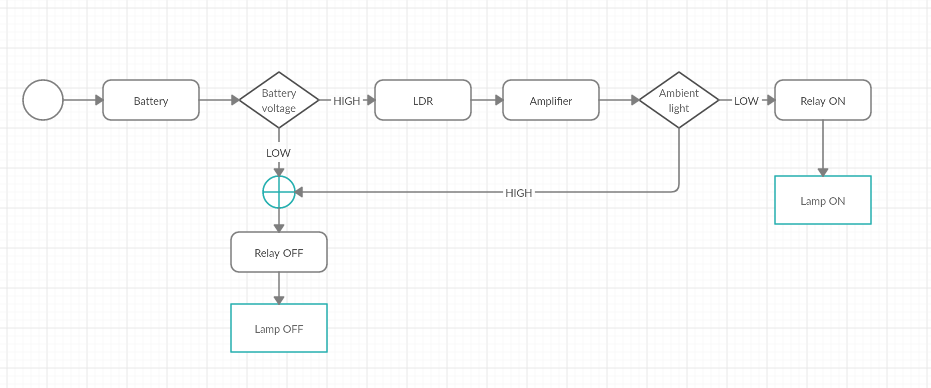


Component diagram

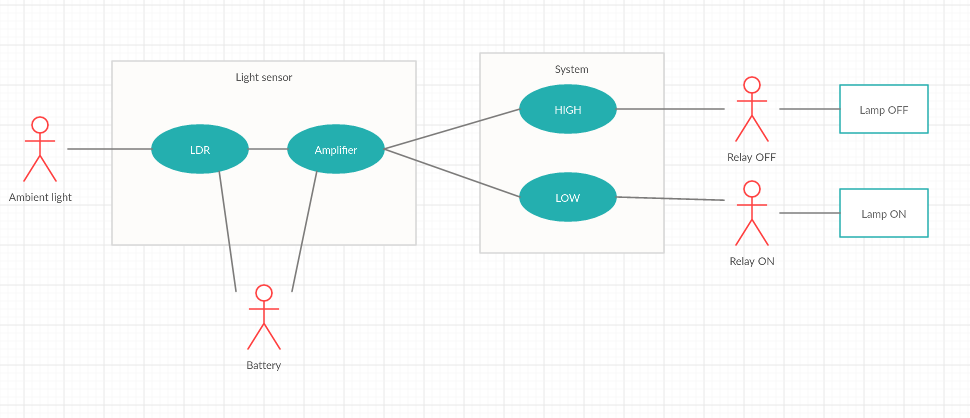


Package diagram

* + Behavioral diagram:



Activity diagram



Use case diagram

## **Activity 2** – Agile Aspects

* User story 1:

As a responsible citizen, I want to conserve energy so that the systems are more sustainable and work for longer periods of time.

* User story 2:

As a daily commuter, I want the street lamps to turn ON whenever it is dark so that the roads are always visible.

* User story 3:

As a daily commuter, I want the street lamps to work appropriately in all weather conditions so that roads are visible.

## **Activity 3**– System/Software Development for Genesis week1 final submission

* **Project Name:** Hopping numbers
* Description:

The Hopping Numbers software is an interactive piece of code that helps the user in identifying if certain numbers are,

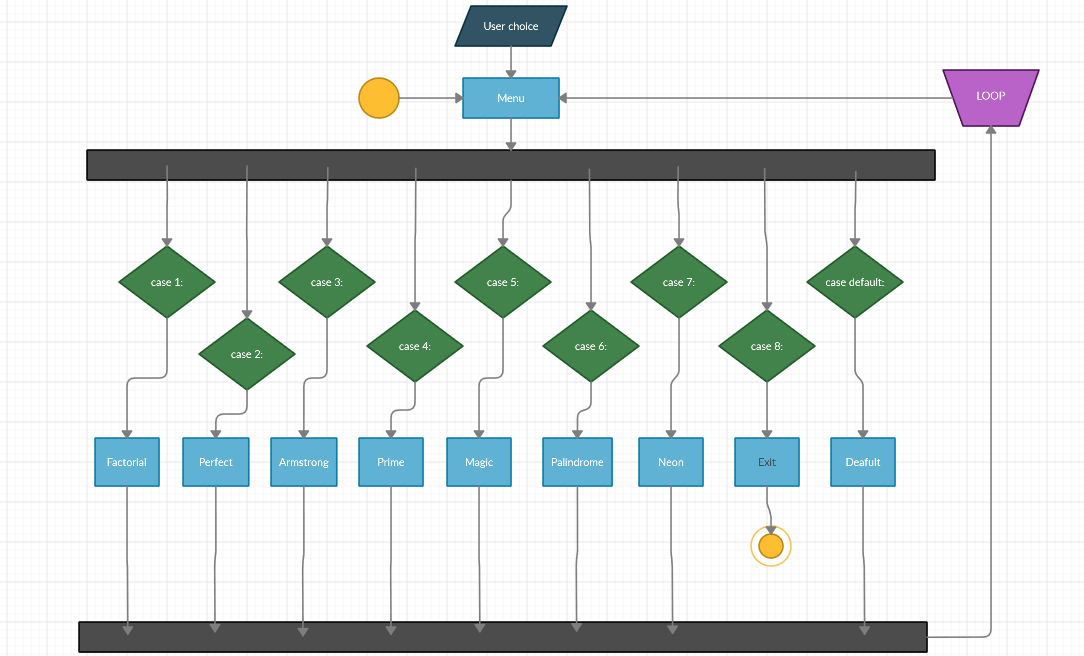
* + Factorial
  + Perfect
  + Armstrong
  + Prime
  + Magic
  + Palindrome
  + Neon
* Requirements:
* High level requirements:

|  |  |
| --- | --- |
| **ID** | **Description** |
| HL\_01 | Must print if the input number is armstrong number or not |
| HL\_02 | Must print the factorial of the input number |
| HL\_03 | Must print if input number is magic number or not |
| HL\_04 | Must print if input number is neon number or not |
| HL\_05 | Must print if input number is perfect number or not |
| HL\_06 | Must print if input number is palindrome number or not |
| HL\_07 | Must print if a number is prime number or not |

* Low level requirements:

|  |  |
| --- | --- |
| **ID** | **Description** |
| LL\_01 | Must work for zero and negative numbers |
| LL\_02 | Must work for zero |
| LL\_03 | Must work for zero and negative numbers |
| LL\_04 | Must work for zero and negative numbers |
| LL\_05 | Must work for zero and negative numbers |
| LL\_06 | Must work for zero and negative numbers |
| LL\_07 | Must work for zero, one and negative numbers |

* Design



Activity Diagram

* Test plan:
  + High level test plan:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Precondition** | **Expected I/P** | **Expected O/P** | **Actual O/P** |
| T\_HL\_01\_1 | Armstrong number check for standard input | User input | 153 | 1 | 1 |
| T\_HL\_01\_2 | Not armstrong number check for standard input | User input | 154 | 0 | 0 |
| T\_HL\_02 | Factorial of a number check for standard input | User input | 5 | 120 | 120 |
| T\_HL\_03\_1 | Magic number check for standard input | User input | 1234 | 1 | 1 |
| T\_HL\_03\_2 | Not magic number check for standard input | User input | 1235 | 0 | 0 |
| T\_HL\_04\_1 | Neon number check for standard input | User input | 9 | 1 | 1 |
| T\_HL\_04\_2 | Not neon number check for standard input | User input | 8 | 0 | 0 |
| T\_HL\_05\_1 | Perfect number check for standard input | User input | 6 | 1 | 1 |
| T\_HL\_05\_2 | Not perfect number check for standard input | User input | 7 | 0 | 0 |
| T\_HL\_06\_1 | Palindrome number check for standard input | User input | 121 | 1 | 1 |
| T\_HL\_06\_2 | Not palindrome number check for standard input | User input | 123 | 0 | 0 |
| T\_HL\_07\_1 | Prime number check for standard input | User input | 37 | 1 | 0 |
| T\_HL\_07\_2 | Not prime number check for standard input | User input | 10 | 0 | 0 |

* + Low level test plan:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Precondition** | **Expected I/P** | **Expected O/P** | **Actual O/P** |
| T\_LL\_01\_1 | Armstrong number check for zero | User input | 0 | 1 | 1 |
| T\_LL\_01\_2 | Armstrong number check for negative number | User input | -153 | 0 | 0 |
| T\_LL\_02 | Factorial of zero | User input | 0 | 1 | 1 |
| T\_LL\_03\_1 | Magic number check for zero | User input | 0 | 0 | 0 |
| T\_LL\_03\_2 | Not magic number | User input | -1234 | 0 | 0 |
| T\_LL\_04\_1 | Neon number check for zero | User input | 0 | 0 | 0 |
| T\_LL\_04\_2 | Neon number check for negative number | User input | -9 | 0 | 0 |
| T\_LL\_05\_1 | Perfect number check for zero | User input | 0 | 0 | 0 |
| T\_LL\_05\_2 | Perfect number check for negative number | User input | -6 | 0 | 0 |
| T\_LL\_06\_1 | Palindrome number check for 0 | User input | 0 | 1 | 1 |
| T\_LL\_06\_2 | Palindrome number check for negative number | User input | -121 | 0 | 0 |
| T\_LL\_07\_1 | Prime number check for zero | User input | 0 | 0 | 0 |
| T\_LL\_07\_2 | Prime number check for one | User input | 1 | 1 | 1 |
| T\_LL\_07\_3 | Prime number check for negative number | User input | -37 | 0 | 0 |

* CI Workflow for C programming (GENESIS):

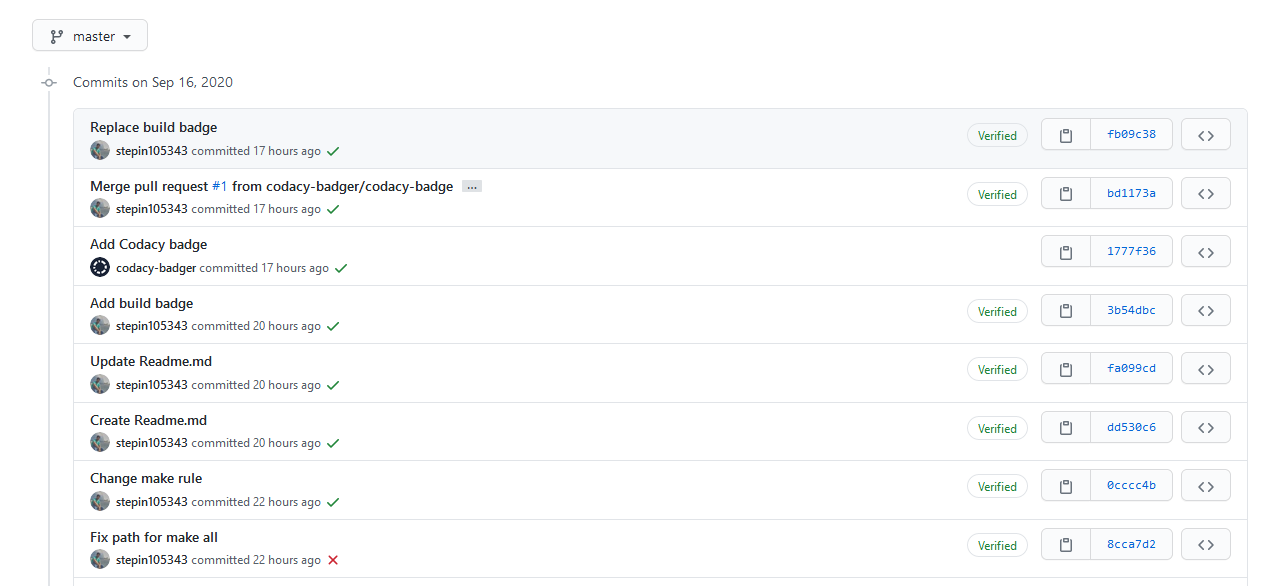
[GENESIS Github Repository](https://github.com/99002670/bunnies)

* **Commit log:**

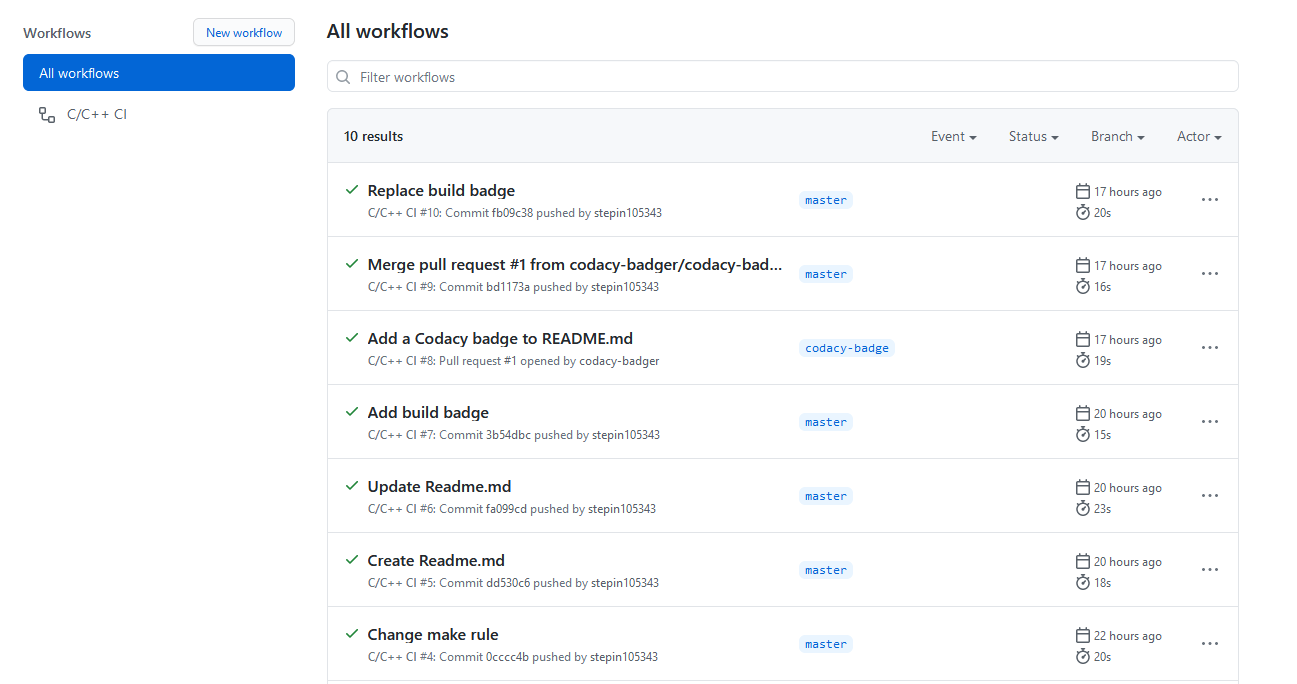
## **Appendix** –CI Workflow for C Programming (STEPin)

[STEPin Github Repository](https://github.com/stepin105343/stepin-practice)

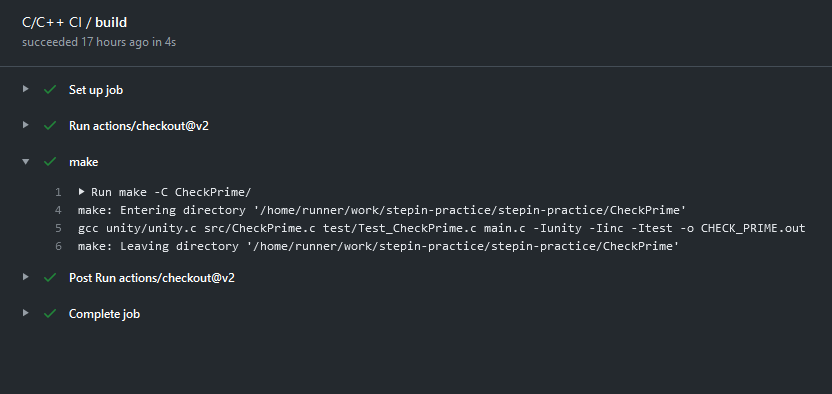
* **Commit log:**



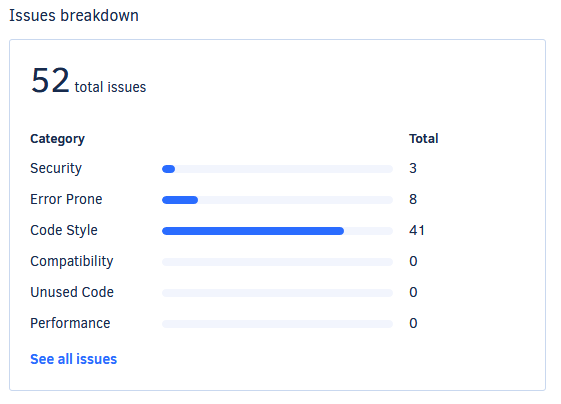
* **Actions:**



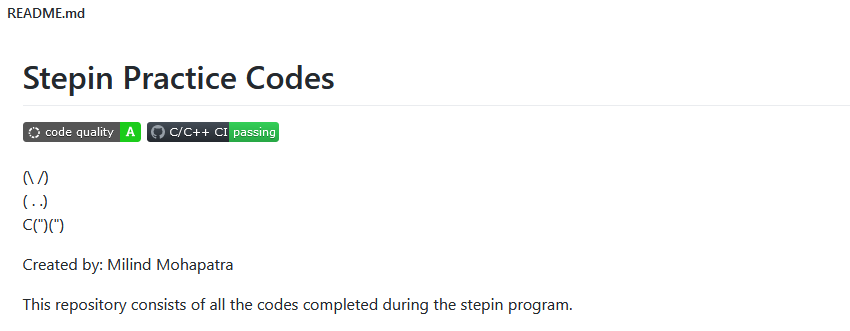
* **Build:**



* **Codacy issues:**



* **Badges:**



* **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