

Learning Report - SDLC



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved By** | **Remarks/Revision Details** |
|  | 15-10-2020 | Praveen Kumar Chaubey | Shweta Melavane  Arvind Kumar | PSP and SK |  |
|  | 16-10-2020 | Praveen Kumar Chaubey | Shweta Melavane  Arvind Kumar | PSP and SK |  |
|  | 17-10-2020 | Praveen Kumar Chaubey | Dhanshree Vinherkar  Abhishek Pradeep | PSP and SK |  |

# 

**Contents**

**TABLE OF CONTENTS.....................................................................................................................................................3**

**LIST OF FIGURES............................................................................................................................................................4**

[**Activity and Tasks**](#_heading=h.3znysh7)

**Individual Activity 1**

**REQUIREMENTS AND RESEARCH...............................................................................................................5**

**PRODUCT SELECTED –** Drinking Water Vending Machine

Ageing

Costing and Gradation

Definition of Product

**REQUIREMENTS........................................................................................................................................6**

High-level Requirements

Low-level Requirements

**SWOT Analysis .........................................................................................................................................6**

**DESIGN ....................................................................................................................................................7**

High Level Diagrams

Low Level Diagrams

**TEST PLAN...............................................................................................................................................12**

**TEST CASES..............................................................................................................................................12**

**Individual Activity2 ………………………………………………………………………………………………………………………..…..18**

**Individual Activity3 ………………………………………………………………………………………………………………………..…..26**

**Individual Activity4………………………………………………………………………………………………………………………..…….29**

**Individual Activity5………………………………………………………………………………………………………………….……………44**

T**eam** [**Activity 1– System/SOFTWARE DEVELOPMENT………………………………………………………………………………………. 5**](#_heading=h.2et92p0)

**Difference between SYSML and UML** …………………………………………………………………………………….12

**Team Activity 2– System/SOFTWARE DEVELOPMENT**………………………………………………………………………………………**….5**

**Impact of Failures in Medical Devices 15**

**Team Activity 3- System/SOFTWARE DEVELOPMENT 5**

**Agile Development and Analysis Team Activity 4- System/SOFTWARE DEVELOPMENT 5**

**Tools Analysis**  **37**

**Team Activity 5- System/SOFTWARE DEVELOPMENT 5**

**Calculator**  50

**REFERENCES………………………………………………………………………………………………………………………………………………………………60**

**LIST OF FIGURES**

Fig.1 - Use Case Diagram for Drinking Water Vending Machine………………………………………………………………………….7

Fig.2 - Sequence Diagram for Drinking Water Vending Machine ……………………………………………………………..………..8

Fig.3 - Activity Diagram for Drinking Water Vending Machine ……………………………………………………………………………9

Fig.4 - Class Diagram for Drinking Water Vending Machine ……..………………………………………………………….…….………10

Fig.5 - Object Diagram for Drinking Water Vending Machine …………………….………………………………………………..….…11

Fig.6 - SySML as subset of UML …………………….………………………………………….………………………………….….…….……..……12

Fig.7 - UML Diagram …………………………………………………………..…………………….……………………………..…………………..….…13

Fig.8 - SYSML Diagram …………………………………………………….…………………….………………………………………………………...…14

[Fig.9 - Use Case Diagram of Mini project 21](#_Toc52744710)

[Fig.10 - Activity Diagram of Mini project](#_Toc52744711) 21

[Fig .11 - Component Diagram of Mini project](#_Toc52744712) 22

[Fig 12 - Class Diagram of Mini project](#_Toc52744713) 23

Fig 13 - Badges in GitHub……………………………………………………………………………………………………………………………………..27

Fig 14 - Issue in GitHub……………………………………………………………………………………………………………………..……………......28

Fig 15 - Ageing and cost ……………………………………………………………………………………………………………………………..……….50

Fig 16 – Use case UML Diagram……………………………………………………………………………………………………………………..…….52

Fig 17 - Activity UML Diagram …………………………………………………………………………………………………………………………….53

Fig 18 - Class UML Diagram ………………………………………………………………………………………………………………………….…….53

Fig 19 - Object UML Diagram …………………………………………………………………………………………………………………….……….54

.

## **Individual Activity 1**

**Research**

**Ageing:**

* Same Technology as coin operated Telephone
* Seaga India, a 100% subsidiary of the Seaga Group of USA, is the pioneer for bringing the concept of vending machines to India.
* In the nineteenth century, the first vending machine to be successfully commercialized by Thomas Adams was used for the sale of their chewing gum in underground stations of New York.
* In 1902 the first company of vending machines emerged, Horn & Hardart Baking Company in Philadelphia.
* In the 50s, the automatic coffee machines expanded to several small businesses.
* In the ‘60s, two manufacturers have introduced machines with the hypothesis of automatic note changers.
* Vending machine market in India is still at a nascent stage and is expected to witness dynamic growth in the near future.
* The market will be boosted by the increasing consumerism coupled with changing lifestyle, resulting in increased demand for vending machines.

**Cost Gradation:**

In present market, there are various water vending machine with various features and as per public survey, there are slight disadvantages because of

* 1. Huge Machines
  2. Initial Investment is high (3-5 lakhs)
  3. Cost of transportation (10 Thousand)
  4. Cost of Maintenance (25 Thousand)

**Proposed Model (Requirements):**

Let’s propose a model with following requirements:

Proposed specifications-

* Comfortable,
* Good chunk of electronics Basic Sensors
* Easy to use
* Efficient Power Management
* Reliable in harsh Condition
* Installation and usage should be cost effective.
* Product should occupy less space.
* Usage of RO safe component.

**HIGH LEVEL REQUIREMENTS**

* Dispense water based on the value of the coin.
* Recognize 6 group of coin in different denominations.

**LOW LEVEL REQUIREMENTS:**

1. Coin Size
2. Working Temperature
3. Storage temperature
4. Buy item
5. Choose item
6. Pay item
7. Take item
8. Recharge item
9. Setup machine
10. Check

**S.W.O.T:**

Strength

1. Wide Usage especially in cities
2. Pay Per Use
3. Ease of Usage
4. Portability

Weakness

1. Initial Investment
2. Size of Product
3. Lack of Practical experiments

Opportunities

1. Migration of Population in cities
2. Have good demand in the market
3. Wide dynamic Market

Threats

1. language barrier
2. Physical threats
3. theft
4. Appearance of Competitions in short time

**DESIGN**

**BEHAVIOURAL DIAGRAMS**

**LOW LEVEL REQUIREMENT**

**USECASE DIAGRAM**

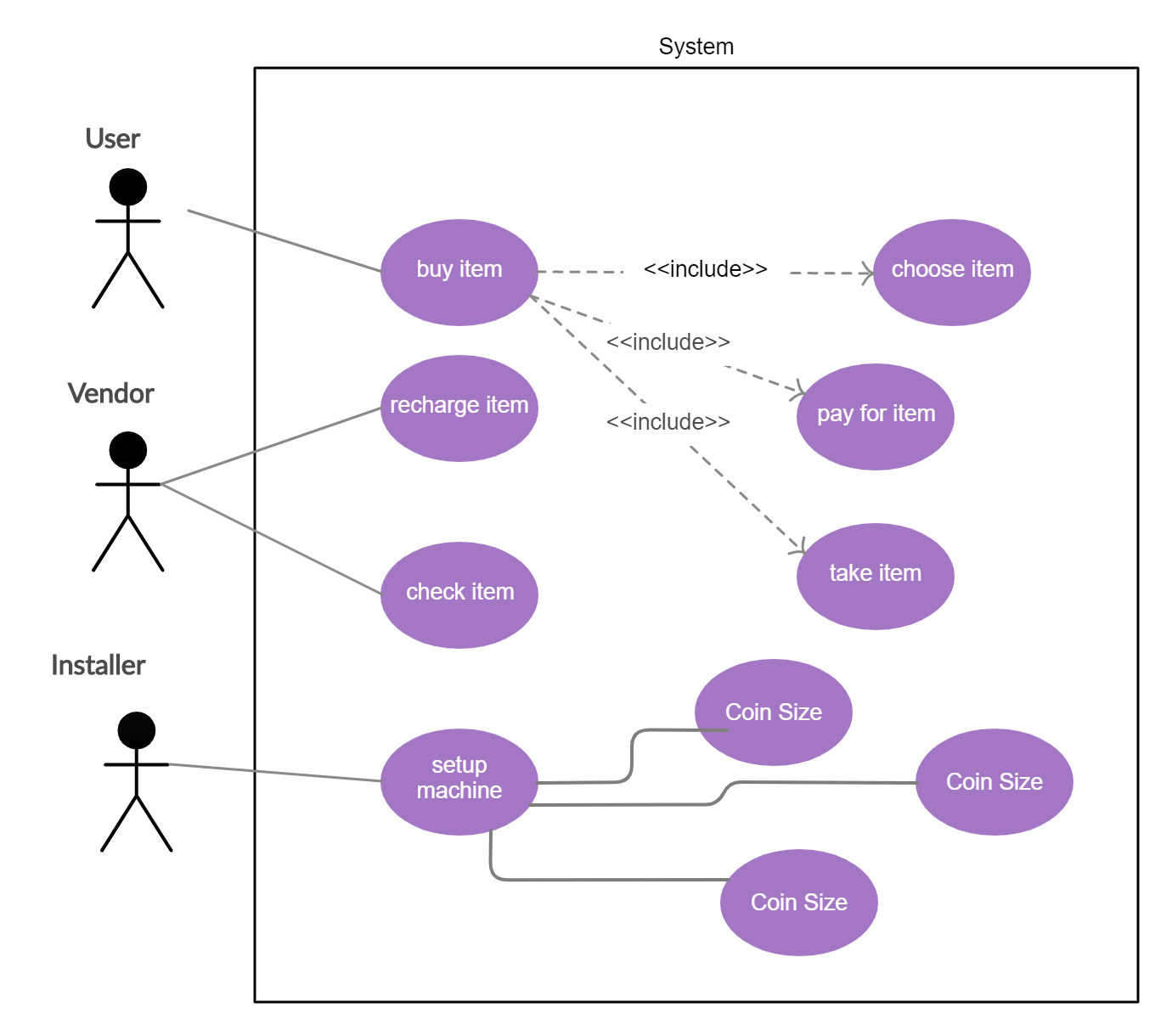


Fig.1 Use Case Diagram

**SEQUENCE DIAGRAM**

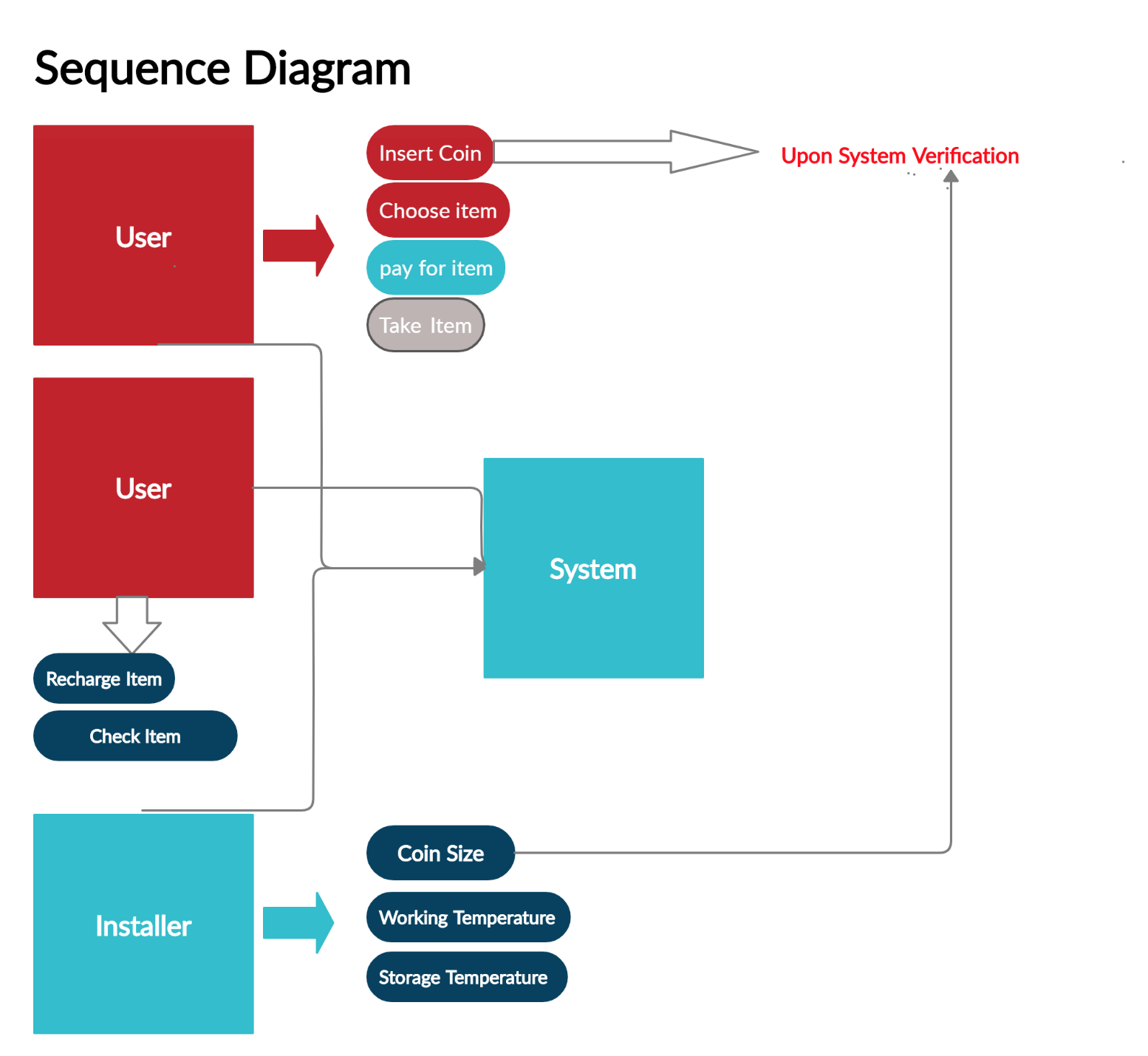


Fig.2 Sequence Diagram

**ACTIVITY DIAGRAM**

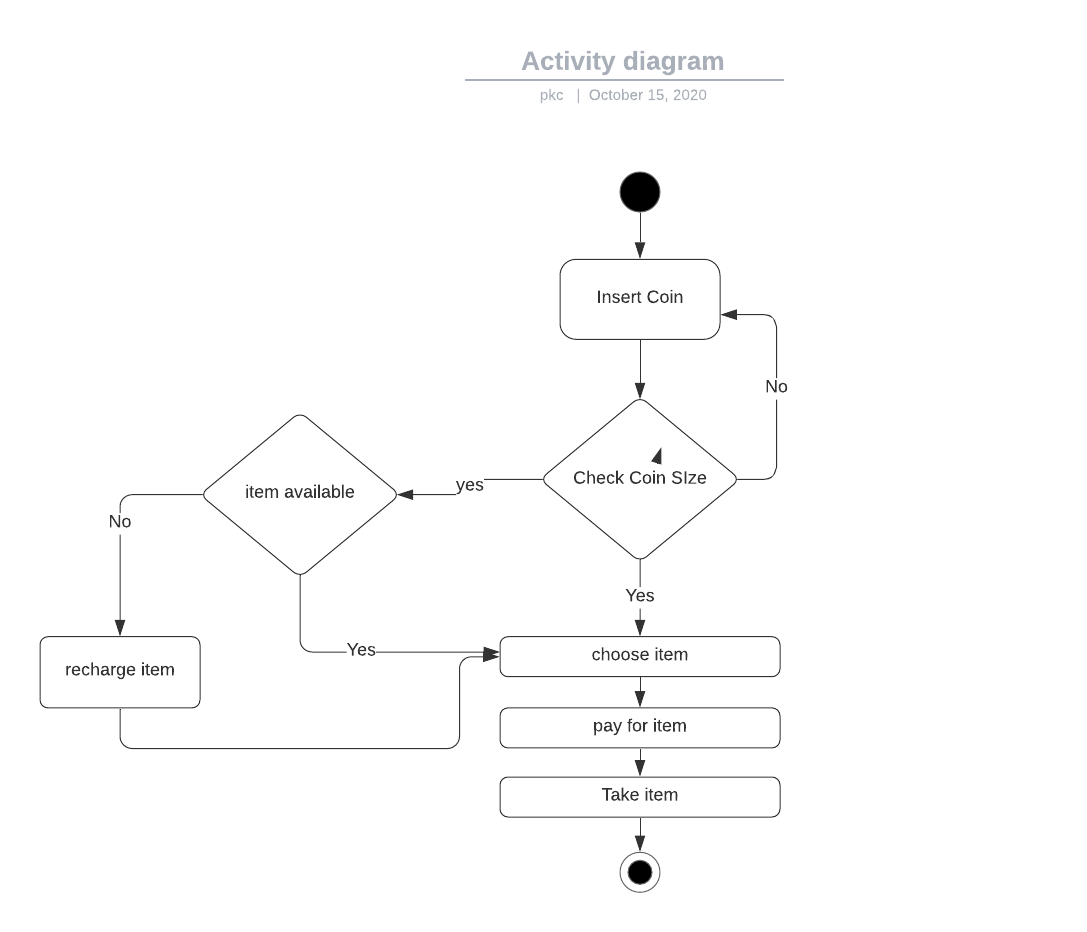


Fig.3 Activity Diagram

**STRUCTURAL DIAGRAMS**

HIGH LEVEL REQUIREMENT

CLASS DIAGRAM

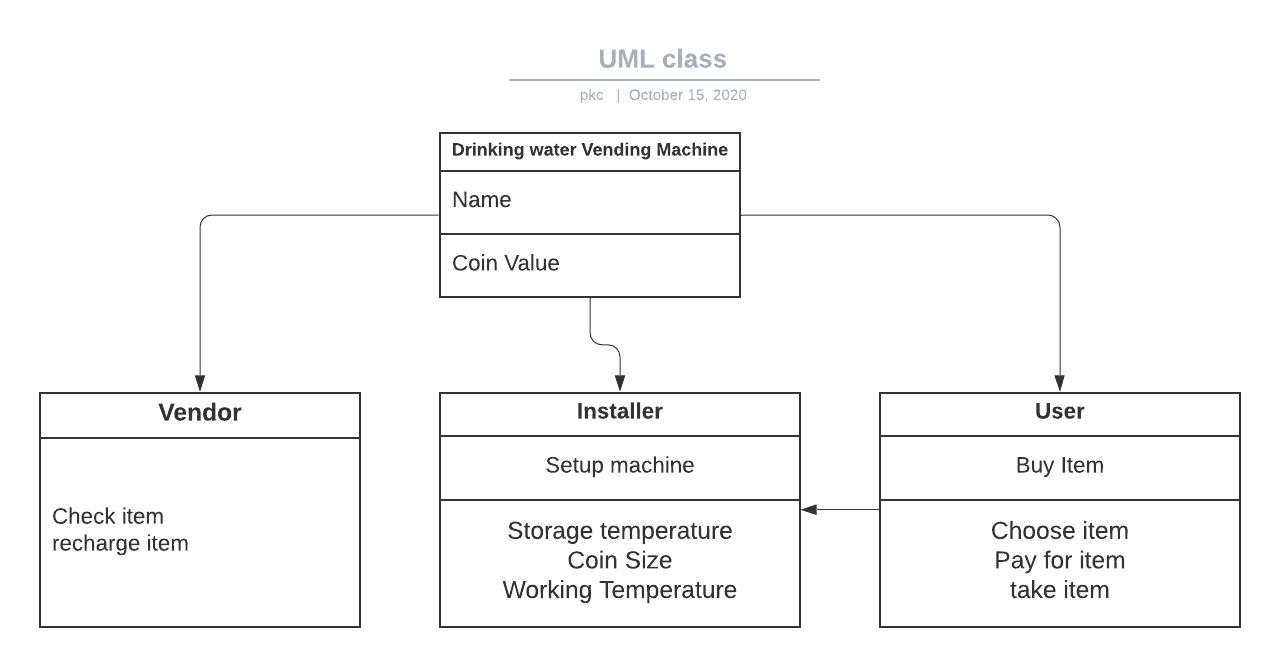


Fig.4 Class Diagram

OBJECT DIAGRAM

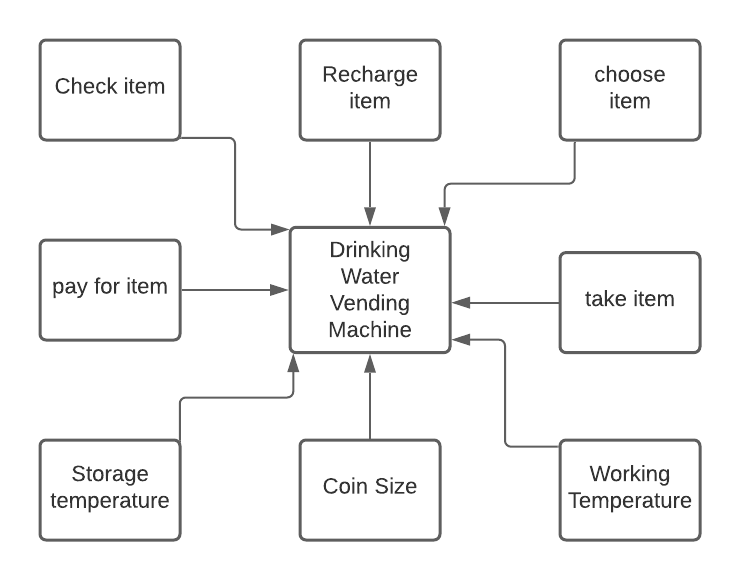


Fig.5 Component Diagram

**TEST PLAN:**

Here we write test cases to verify whether our requirements are fulfilled.

Things we are going to test on this prototype:

1. Value of Coin

2. Liters of water dispensed

3. Working temperature

4. Storage temperate

5. Refilling

6. Checking

The test plan as follows:

Unit testing -> Integration testing.

Unit testing

We are going to test each part before assembling it to its component to ensure that every part is working properly which we cannot identify the defect when it is assembled.

Integration Testing

Testing the components which are assembled and considering many possible ways which it can withstand

**TEST CASES:**

High-level Requirements

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | **ID** | **DESCRIPTION** | **PRE-CONDITION** | **EXPECTED INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** |
| Buy | 1 | Purchase of water | ! empty container | Coin | Water bottle | Water bottle |
| Setup machine | 2 | Installation of device | Defined temperature | Need of product | Successful installation | Successful installation |
| refill | 3 | Refilling of container when its empty | empty container | Coin | Filled container | Filled container |

Low-level Requirements

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | **ID** | **DESCRIPTION** | **PRE-CONDITION** | **EXPECTED INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** |
| Value off coin | 1 | To know value of coin entered | Need of water bottle | Input of Coin | Input to next process | Choose item |
| temperature | 2 | To know working and storage temperature | Null | Data from temperature sensor | Temperature data | Temperature data |

## 

## **Team Activity 1**

DIFFERENCE BETWEEN SYSML and UML

UML is a standardized language for specifying software systems. SySML uses a subset of the diagrams defined by UML and has extensions. It is a UML-profile. While UML is software-centric, SySML is more engineering systems-oriented. For example, SySML is used in system level design of System on Chips. SySML as subset of UML is shown below.

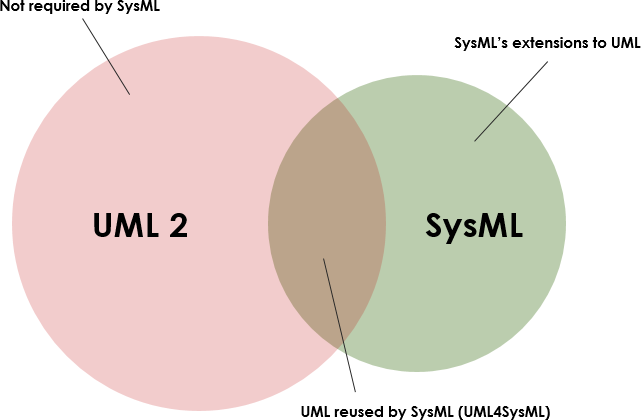


Fig.6

*UML4SysML:*

* Sequence diagram
* State diagram
* Use case diagram
* Diagram of activities
* Package diagram
* Class diagrams and composite structure (used for block definition and internal block diagrams – BDD & IDB)

*SysML Extensions:*

* Definitions for Block Definition and Internal Block Diagrams – BDD & IDB
* Changes in the activity diagram
* Requirements diagram – New
* Parametric diagram – New
* Allocations (traceability) – New

UML has 14 diagrams – 6 structural and 7 behavioral, it is shown below

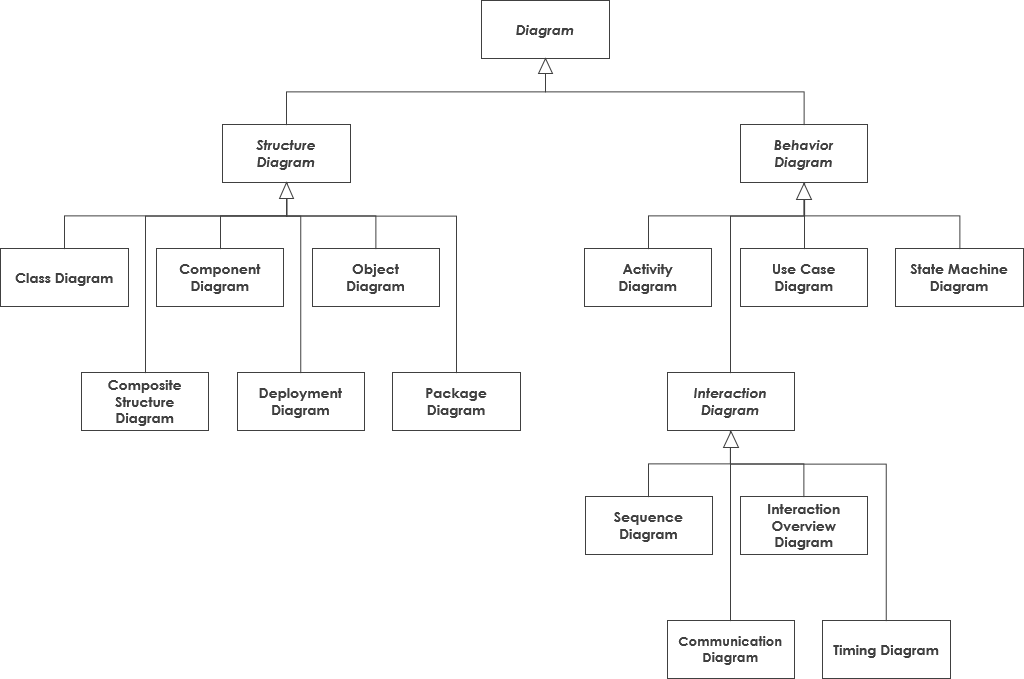


Fig.7 UML diagram

Whereas, SySML has 9 diagrams - 4 structural, 4 behavioral and 1 requirement diagram

* 1. Structural
* The Block Definition Diagram (BDD) replaces the Class Diagram
* The Internal Block Diagram (IBD) replaces the composite structure diagram
* The package diagram remains unchanged
* The parametric diagram is a SysML extension for the analysis of critical system parameters
  1. Dynamic
* The activity diagram is slightly modified for SysML
* Sequence, state, and use case diagrams remain unchanged
  1. The requirements diagram is a SysML extension

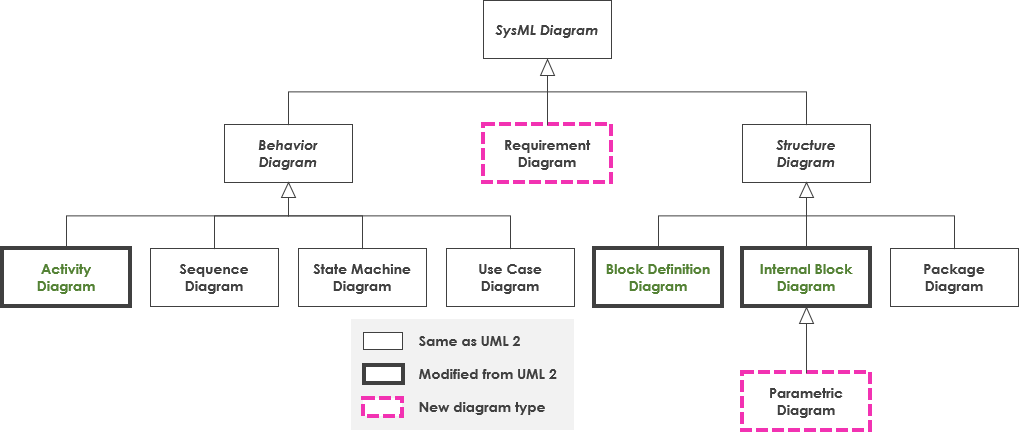


Fig.8 SySML Diagram

## **Team Activity 2**

## **Impact of failures in Medical Devices**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr No.** | **Product** | **Cause** | **Impact** | **Outcome** | **Reference** |
| 1 | Philips Respironics Ventilators with Defective Components | Power Management Board issue | Caused the ventilators to not actually deliver mechanical breaths to the patients | All the products were recalled and the customer hospitals were provided with replacements for the same | <https://www.nsmedicaldevices.com/analysis/medical-device-failures-fda-recalls/> |
| 2 | Medtronic insulin pumps | cyber security risk | Unauthorized person could then instruct the pump to either over-deliver insulin to a patient, leading to low blood sugar (hypoglycemia), or stop insulin delivery, leading to high blood sugar, diabetic ketoacidosis, and even death — according to the FDA. | US medical device company was forced to issue a recall of more than 1,000 of these devices. | https://www.nsmedicaldevices.com/analysis/medical-device-cyber-security/ |
| 3 | GE Healthcare Infant Resuscitators with Assembly Error | Resuscitators could limit the supply of oxygen and air wall inlet fittings on their back panels were reversed during assembly | Potentially injure or kill infants during the oxygen supply | Recall of the manufactured products to prevent the casualities. | <https://www.nsmedicaldevices.com/analysis/medical-device-failures-fda-recalls/> |
| 4 | Physio-Control defibrillators | it was discovered that electrical problems with the device may prevent it delivering the charged shock needed to revive a patient in cardiac arrest. | In response, Physio-Control told its customers to perform a “user test” every day – and to contact the company immediately if the device failed this test, to arrange for the device to be repaired. | The FDA said a delay in delivering the shock could result in permanent organ damage, brain injury, or death. | <https://www.nsmedicaldevices.com/analysis/medical-device-failures-fda-recalls/> |
| 5 | Duodenoscopes | Duodenoscopes, with their many small working parts, are notoriously difficult to sterilise after use. | Infections from duodenoscopes caused three deaths and 45 infections in a six-month period. | Recent tests performed by duodenoscope manufacturers Fujifilm, Pentax and Olympus at the FDA’s behest found that one in 20 duodenoscopes retained particles of diseases such as E. coli and pseudomonas aeruginosa. | <https://medical-technology.h5mag.com/medical_technology_nov19/duodenoscopes_a_dirty_problem> |

## **Mini project**

**Project Theme: Library Administration**

**1.1 Research:**

**1.1.1 Ageing:**

**Pre-Medieval India**

1. Vedic age instructions were imparted “orally, without the medium of books.”(Agarwal, 1954) Taxila from 700 B.C. to 300 A.D. was considered to be the most respected seat of higher learning and education in India.

2. In 400 A.D., there came into being one of the biggest known universities, the Nalanda University, which by 450 A.D. became a renowned seat of learning.

**Medieval India**

The existence of academic libraries during the medieval period of Indian history is not known, though the Muslim rulers did patronize libraries in their own palaces. A lone exception, however, was a library attached to a college at Bidar, (Gawan,1463-82) having a collection of 3000 books on different subjects.

**Modern India**

1. the Montague-Chelmsford reforms of 1919, the Government of India Act of 1935, and the Sargent Committee Report etc. laid foundation for establishment of libraries in various parts of the country.

2. The most comprehensive and significant document on the university and college libraries is the Report of the UGC library committee, chaired by Ranganathan of 1957.

3. The National Knowledge Commission was set up by the Government of India on 13th June 2005.

**1.2 Description of Solution:**

Library Administration mainly comprises of 4 different department.

1. Library Books Management

It consist of information regarding books like bookname, publish year, Book ISBN number, author name, books which can be borrowed etc.

2. Library Members Management

This department consist of information regarding Members of library such as their name, joining year, books issued, borrowed book, removed book etc.

3. Library Store Management

This department consist of functionalities like finding members and books by id, finding books which exist in the library and books which has been issued.

4. Utilities provided

This is user interface, It helps users to communicate with backend process.

C++ Concepts used:

* Inheritance
* Classes and objects
* Vectors and list
* strings
* constructors and destructors etc.
* threads

Linux Concepts used:

* Make files for the execution of the project.
* Static and Dynamic libraries for the faster execution and quicker run time.

**1.3 SWOT Analysis:**



**1.4 Requirements:**

**1.4.1 High Level Requirements:**

|  |  |
| --- | --- |
| ID | Description |
| H1 | Management of Book Information |
| H2 | Management of Library Member Information |
| H3 | Store Management |
| H4 | Provide library Utilities to Members |

**1.4.2 Low Level Requirements:**

|  |  |
| --- | --- |
| ID | Description |
| L1 | Get the information about book |
| L2 | List out books which can be borrowed to member. |
| L3 | Set the information about book. |
| L4 | Get Member information |
| L5 | Set Member Information |
| L6 | Get Member borrowed books count |
| L7 | Check whether Member has borrowed books |
| L8 | Find Member by id |
| L9 | Print New Member information |
| L10 | Print Borrowed book information |

**1.5 Design:**

**1.5.1 High Level Design:**

**Behavioral Diagram:**

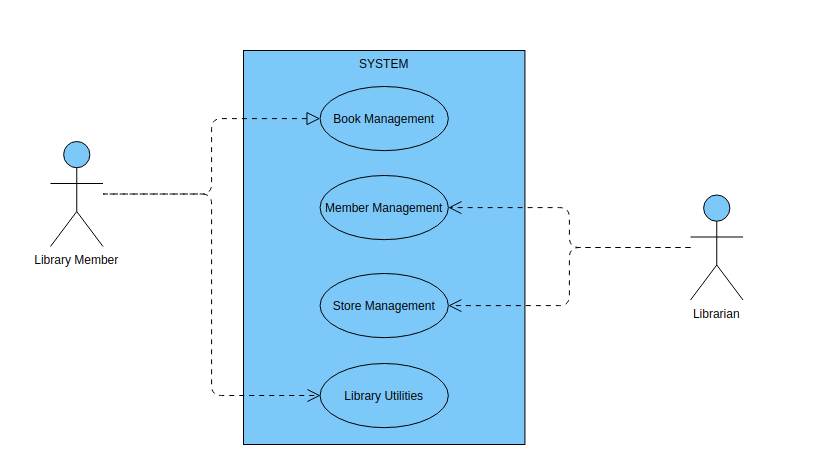


Fig 9. Use Case Diagram

**Structural Diagram:**

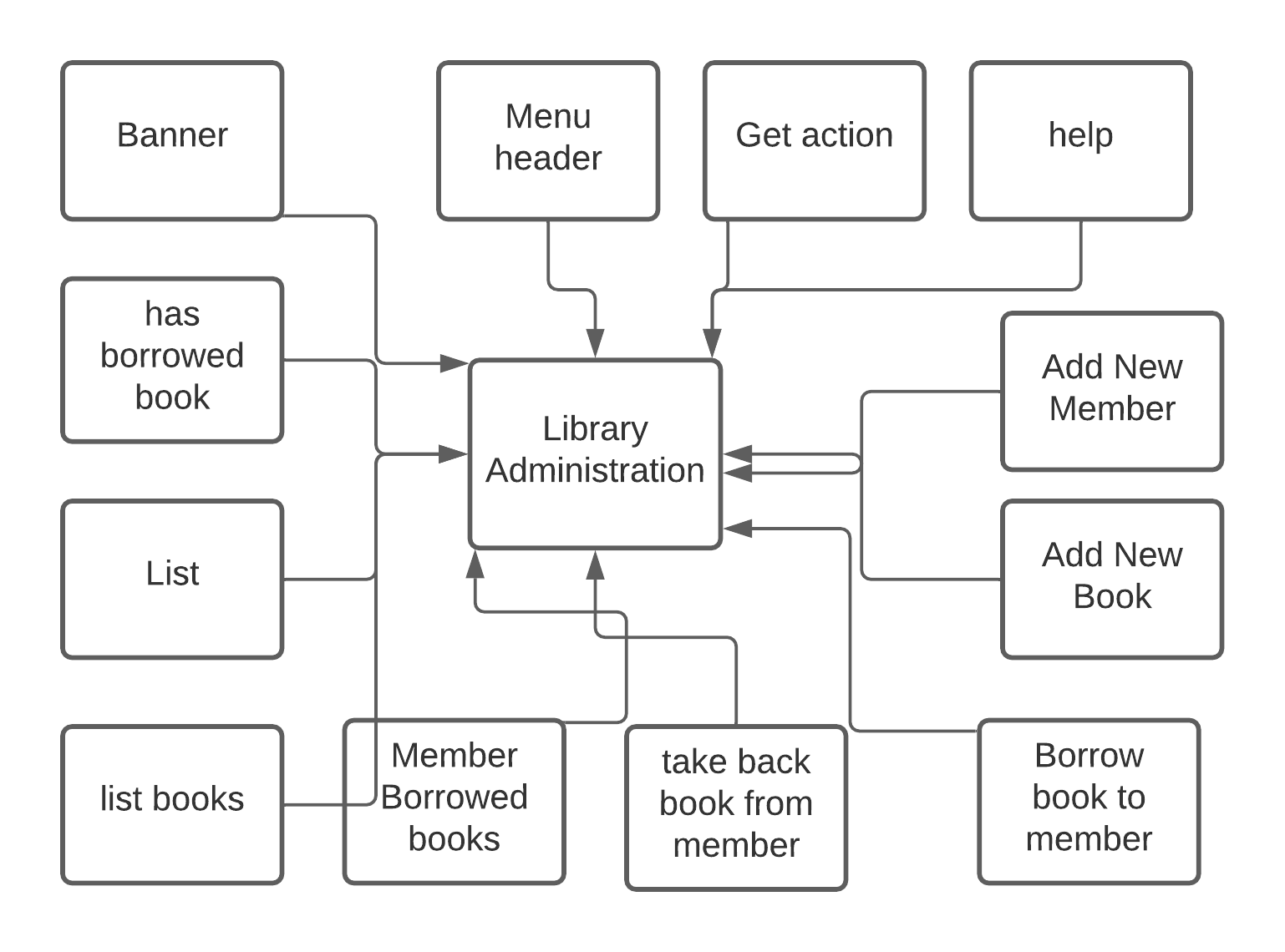
****

Fig 10 Component Diagram

**1.5.2 Low Level Design:**

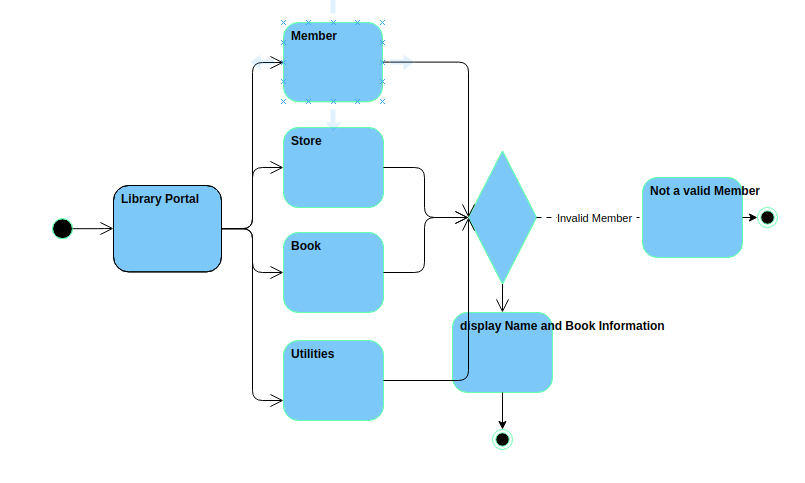
**Behavioral Diagram:**

Fig 11. Activity Diagram

**Structural Diagram:**

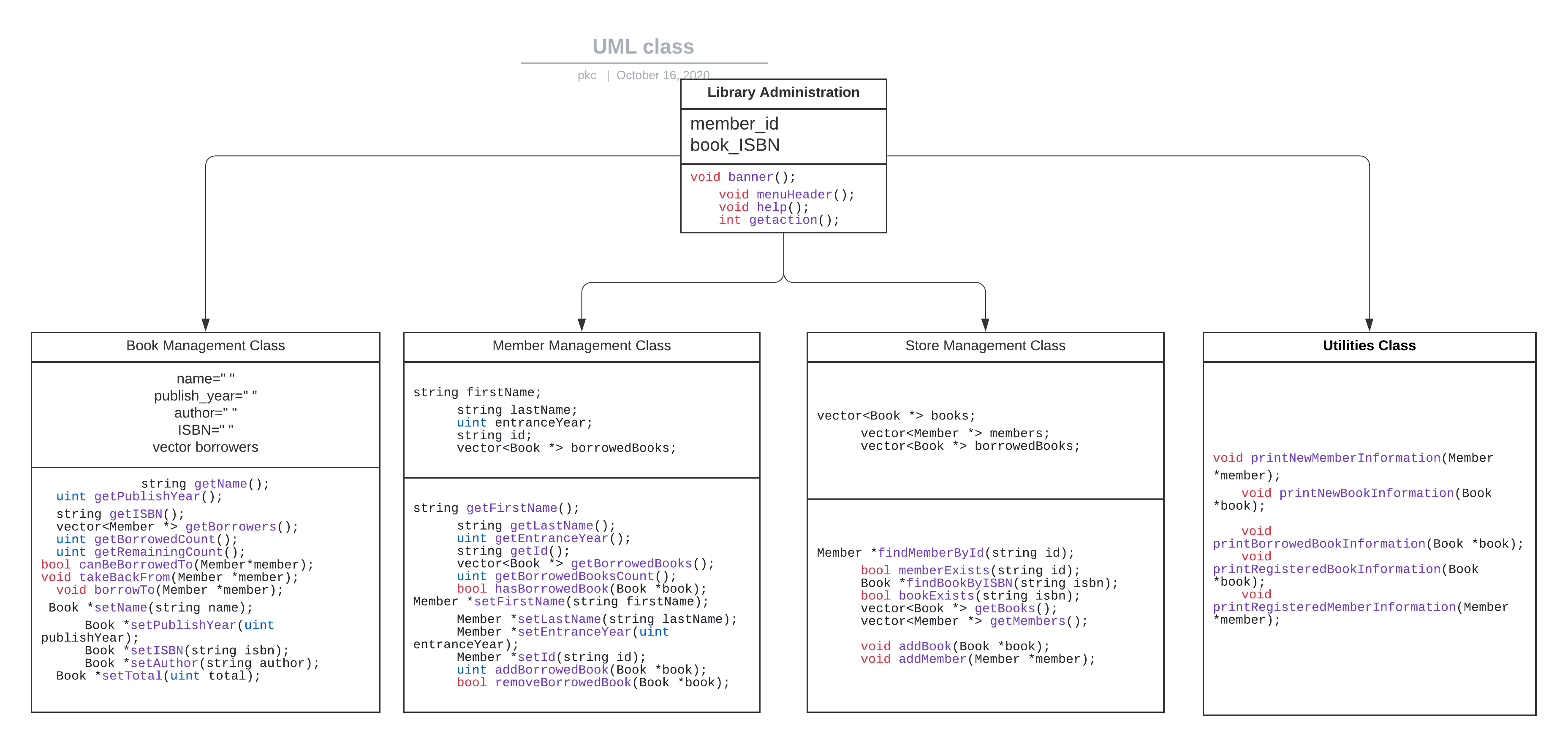


Fig 12. Class Diagram

**1.6 Test Plan:**

**1.6.1 High Level Test Plan (Integration Test Plan)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Precondition** | **Expected IO** | **Expected OP** | **Actual OP** |
| H1 | Management of Library book information | Set name of author,ISBN,book,publish year etc | Set using inputs to the console | Successful set of Book details | Successful set of Book details |
| H2 | Management of Library Member Information | Set name of member, entrance year, id, borrowed books | Set using inputs to the console | Successful set of Member details | Successful set of Member details |
| H3 | Finding member and books using id | Get the id and ISBN set from the user | Set using inputs to the console | Find member and books using id and ISBN | Find member and books using id and ISBN |
| H4 | Provide library Utilities to Members | Get the command of utilities from user | Get the command of utilities from user | Successful completion of user command | Successful completion of user command |

**1.6.2 Low Level Test Plan (Unit Test Plan)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Precondition** | **Expected IO** | **Expected OP** | **Actual OP** |
| L1 | Get the information about book | Set ISBN and author name | Set using input Console | Successful information about book | Successful information about book |
| L2 | List out books which can be borrowed to member. | Enter member id | Set using input Console | Members remaining book count | Members remaining book count |
| L3 | Set the information about book. | Name of Book, Author, ISBN, publish year | Set using input Console | Successful setting of book information | Successful setting of book information |
| L4 | Get Member information | Enter member id | Set using input Console | Successful information about Member | Successful information about Member |
| L5 | Set Member Information | Name, entrance year, id, Borrowed book list | Set using input Console | Successful setting of Member information | Successful setting of Member information |
| L6 | Get Member borrowed books count | Get Borrowed list data from book class using ISBN | Set using input Console | Get the number of books member has borrowed | Get the number of books member has borrowed |
| L7 | Check whether Member has borrowed books | Get ISBN of borrowed books | Set using input Console | Return true if ISBN matches | Return true if ISBN matches |
| L8 | Find Member by id | Member ID | Set using input Console | True if ID matches | True if ID matches |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| L9 | Print New Member information | Member name,id,entrance year, | Set using input Console | Display Member Information | BDisplay Member Information |
| L10 | Print Borrowed book information | Get book name,author,ISBN, publish year and book count | Set using input Console | Display Borrowed book information | Display Borrowed book information |

**1.6 Implementation:**

**Repository Link:** [**Mini Project Repo.**](https://github.com/99002503/Mini_Project)

CI framework:

* GIT: Used git for storing and continuous evaluation of the code. Very important tool for Projects
* Build and Make –

Used GitHub Workflow files for GitHub actions namely,

1. C/C++ CI workflow
2. Cppcheck
3. Valgrind
4. Build

* Code Quality: Used CODACY for code quality. “B” Grade for the code implemented in the repository.

## **Individual Activity 3**

Using V-Model Analysis, various functionalities were added in the design and then that was implemented by adding its code to the source code along with its unit testing i.e. after 4 seconds system goes to sleep mode functionality was given to the source code using agile method and based on the user experience and feedback, various input color was added in the source code which enhanced the user interface feature in the console.

Below is the screenshot of GitHub which shows badges of git build framework, code quality, val grind, Cppcheck. Here user is verified using PS no which can be seen clearly in given screenshot.

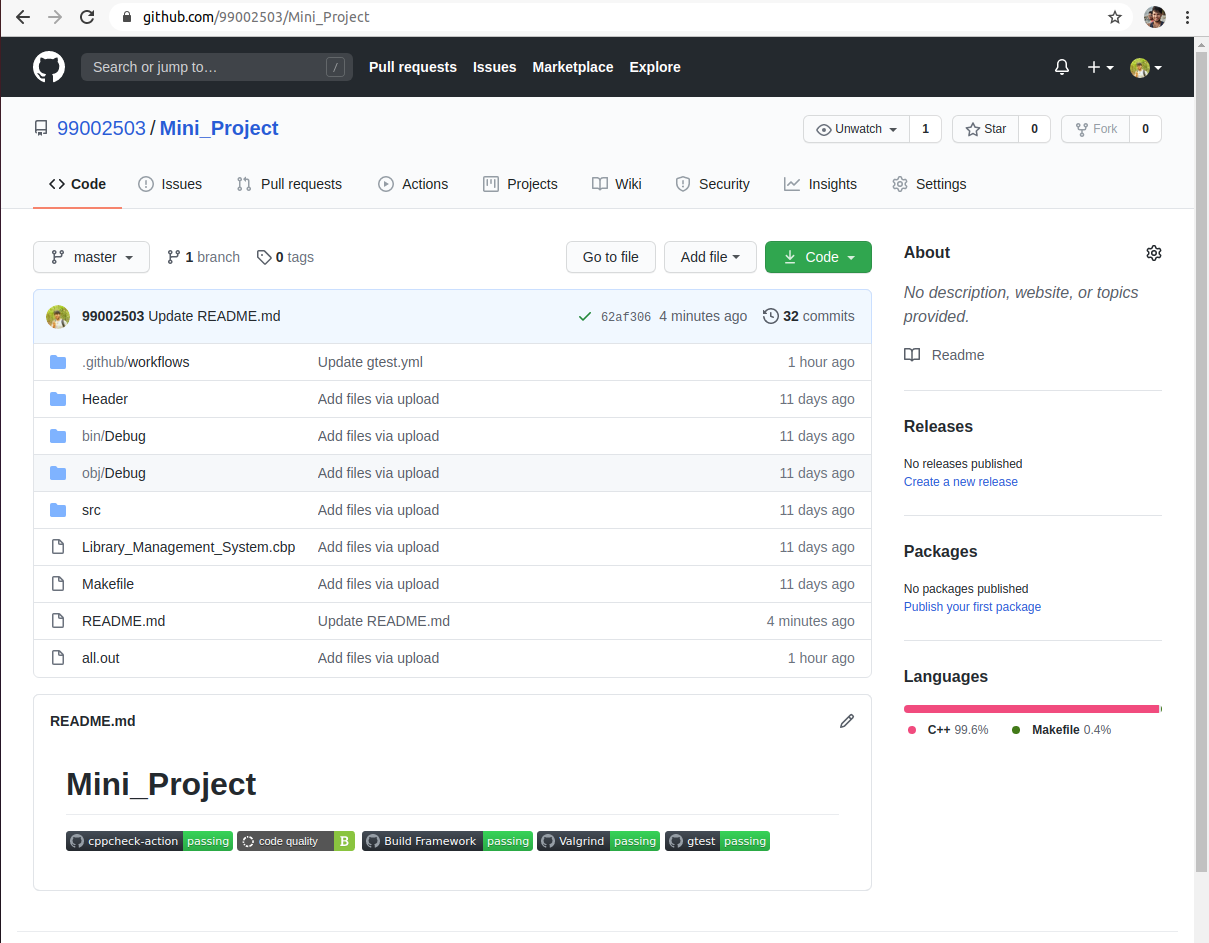


Fig 13. Badges GitHub screenshot

Below screenshot from PS number 99002503 explains about issue regarding code quality raised in the GitHub and later it was closed.

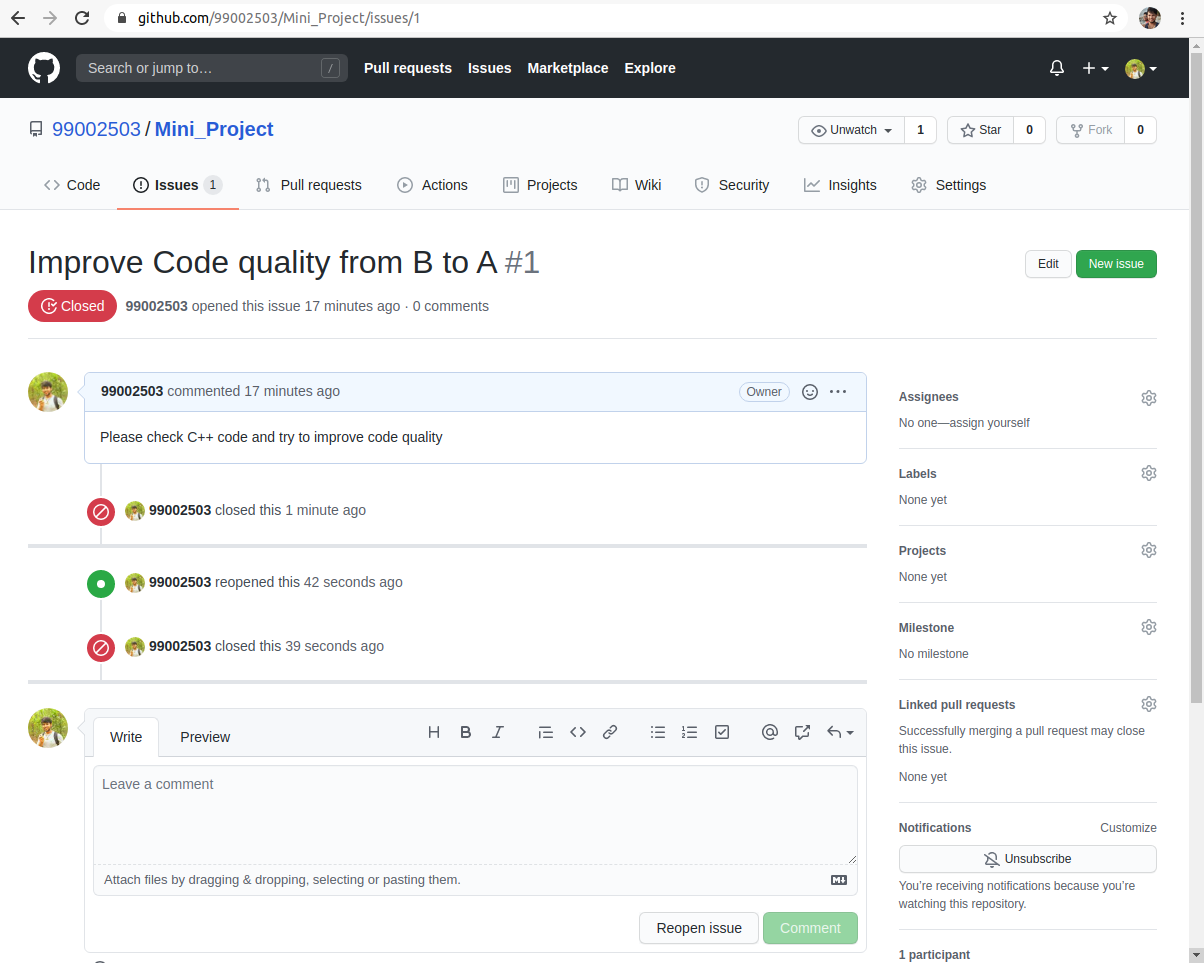


Fig 14. Issue in GitHub

## **Individual Activity 4:**

**Agile Model – Drinking Water Vending Machine**

**Theme:**

Water Vending/Dispensing Machine is an equipment where the customer can use the machine to purchase water bottle for different coin value according to the customer requirements directly through the machine or through the mobile application provided to order. It makes easy to organize, sell and earn profit 24/7, especially without or minimal manpower.

**Epic:**

* Customer can order directly through machine or through mobile application, with available value of coin or through online.
* Customer can view different cost of water bottle and Toppings available.
* Customer can view all the available water bottle and brands through the glass mirror.
* The owner can instantly view the sales statistics 24\*7.
* The owner/supervisor is indicated whenever the machine is low in quantity of water bottles.
* Equip Power Backup to overcome sudden power failures.
* LCD display to place orders. Price of each water bottles is displayed.

**User Stories:**

|  |  |  |
| --- | --- | --- |
| User Type | Epic | User Story |
| Customer | View the different types of water bottle available | The customer should be able to view all the type of Water bottle available through the machine and the price of each Water bottle. |
| Supervisor | The machine should alert when less in water bottle quantity | The Supervisor who is in charge for refilling should be indicated or alarmed whenever the machine less in water bottle quantity and the supervisor can refill the machine on time. |
| Owner | Instant sales statistics 24\*7 | The owner should be able to access all the sales statistics at the instant through his mobile application whenever required by the owner. The total number of water bottle sold and sales generated can be accessed by the owner through his mobile application. |
| Customer | Choose the item and make payment | Whenever the customer chooses the item and payment is made the machine starts the dispensing water bottle and deliver to the customer as soon as possible. |
| Supervisor | Updating the price list | The supervisor can update the price list of the water bottle whenever required. |

## **Team Activity 3:**

**Agile Manifesto**

The developers, who developed a manifesto, called themselves the[Agile Alliance](https://www.agilealliance.org/). They were seeking a restructuring of the traditional software development processes that they saw as cumbersome, unresponsive and too keen and focused on documentation. They wanted to bring a balance and improve on client involvement. They were neither against the documentation nor favored it, as much as we used to, in traditional methods, which was never maintained and reused. They wanted to plan and limit the documentation to the required information. Their new approach to software development was “by doing it and helping others do it.”

To make their view and priority clear, they created this document and called it “Agile Manifesto.”

 When they built this document, they had no idea how quickly their ideas would spread beyond their industry, and in today’s date, the Agile Manifesto is accepted globally.

The Agile Manifesto is a proclamation that articulates four fundamental values and twelve principles that its authors believe;

**Development of Agile Manifesto:**

The Agile software development approach is defined by its commitment to developing software periodically and incrementally. The Agile approach offers the user a new version of software development which is done by following brief periods of work as per commitment and these periods are called **Sprints**.

The Agile manifesto is all about giving the preference to below four factors:

*1.People*

*2.Product*

*3.Communication and*

*4.Responsiveness*

**Agile Manifesto Purpose**

The four values in Agile manifesto promote a software development process, which focuses on quality. They do it by creating software that meets customer’s expectations and needs.

Agile focuses on exactly what is required by the customer; it does not get into making plans and plots; it just goes with the flow and works on small tasks.

Agile is not about “you tell me everything you want,” which is called requirements gathering; this step can take months to document it. Only after documentation, you will be able to start working on software development, and you will get a working product at the last step. The problem here will be “making a change.”

Agile values and principles make it different from others. They make Agile work with change. We ask the client what do they want first and give them the first product, and if any further changes are required, those can be accommodated in the next iteration.

It’s like your food truck would start working quickly rather than waiting for months and make a plan to start a big food chain.

Purpose of the twelve principles is to create and support a work environment which is:

Focused on and around the customer

Aligns to business objectives

Responds and pivots quickly as user needs and market forces change

Agile projects are customer focused and encourage customer guidance and participation. The Agile intends to align development with the business needs, and the success of Agile is apparent.

## **Understanding Ceremonies in Scrum Cycle**

Each Scrum Project could have multiple Release Cycles and each release could have multiple sprints. There are a number of repeating sequence of meetings, to be held before, within and after the sprint cycle.

Scrum suggests three roles: the team, [Scrum Master](https://www.visual-paradigm.com/scrum/what-is-scrum-master/), and [product owner](https://www.visual-paradigm.com/scrum/what-is-project-owner-role-in-scrum/); four ceremonies: the [sprint planning meeting](https://www.visual-paradigm.com/scrum/what-is-sprint-planning/), [Daily Scrum](https://www.visual-paradigm.com/scrum/daily-scrum-meeting-quick-guide/), [sprint review meeting](https://www.visual-paradigm.com/scrum/what-is-sprint-review/), and [sprint retrospective meeting](https://www.visual-paradigm.com/scrum/what-is-sprint-retrospective-meeting/); and three artifacts: the product increment, [product backlog](https://www.visual-paradigm.com/scrum/what-is-product-backlog-in-scrum/), and [sprint backlog](https://www.visual-paradigm.com/scrum/what-is-sprint-backlog-in-scrum/).

**The Agile Scrum Framework**

Meetings or “ceremonies” are an important part of agile development. They help to disseminate timely information, bring common goal and vision, and share team progress to all team members. The complete [Scrum team](https://www.visual-paradigm.com/scrum/what-is-scrum-team/) attends all the ceremonies except the retrospective, which the product owner may or may not attend. Scrum calls for ceremonies mentioned above that forms structure to each sprint:

**Sprint Backlog Refinement**

[Product Backlog refinement](https://www.visual-paradigm.com/scrum/what-is-product-backlog-refinement/) is the act of adding detail, estimates, and order to items in the Product Backlog. This is an ongoing process in which the Product Owner and the Development Team collaborate on the details of Product Backlog items. During Product Backlog refinement, items are reviewed and revised.

As mentioned above, Product Backlog refinement is an ongoing activity, and *unless it is being conducted at scale it does not restricted to be a time-boxed event (or meeting)*. However, there is nothing to stop teams from time-boxing each refinement session anyway. In general, it is good practice to use time-boxing.

* **Sprint Planning Meeting**

The goal of Sprint Planning is to answer the questions “What are we going to work on, and how are we going to do it?” It’s also important for the team to have a shared goal and a shared commitment to this goal before beginning their Sprint – the list of items the team plans to work on during that specific Sprint. The team then breaks down these items into tasks, typically no bigger than a 2 days’ worth of work.

* **Daily Stand-up Meeting**

Once we begin a Sprint, we have what we call a Daily Scrum every day. Organized by the Scrum Master, Daily Scrum is typically a 15-minute stand-up meeting to synchronize the work of team members, i.e. what’s done on the prior day, what needs to be done today, identify any impediments, and creates visibility around the work that everyone is doing in the Sprint. At the end of the Sprint, the goal is to have a Potentially Shippable Product Increment (PSPI). We’re trying to get something of incremental value done every Sprint.

* **Sprint Review Meeting**

Held at the end of each sprint to demonstrate the added functionality. The goal is to get feedback from the product owner and other stakeholders to ensure that the delivered increment met the business need and to revise the Product Backlog based on the feedback. This feedback will then become items that will be looped back into the Product Backlog, where it can be ordered and pulled in by the team in a future Sprint.

* **Sprint Retrospective Meeting**

Retrospectives typically last 90 minutes and are there to help us incorporate continuous improvement into our team culture and into our Sprint cadence. This is where the Scrum Team meets to reflect on their previous Sprint and to figure out how to improve as a team by asking – what went well, what did not and what can be improved. It allows the team to focus on its overall performance and identify strategies for continuous improvement.

**THEME**

The main aim of the project is to develop a Candy vending Machine for the users to get different Candies.

EPIC

Quantity of Candy

Quality of Candy

Internet availability for Payment and Display on screen for Different Payment types

Self-Diagnosis of Machine

Temperature **range** control

USER STORY

|  |  |  |
| --- | --- | --- |
| User Type | Epic | User Story |
| Buyer | Internet availability for Payment and Display on screen for Different Payment types | When buyer goes to the machine they should see the type of payment i.e.. Offline and online. If online Then see the internet |
| Seller | Self-Diagnosis of Machine | The seller has to make sure that the machine rests the process if the vender sensor detects defect |
| Seller | Temperature range control | The seller has to make sure that there is temperature range control of 14 ‘C to 25’C |
| Buyer | Quantity of Candy | The buyer has to get then exact amount of the candies. |

**ROLES:**

Team Lead, Scrum Master (Scrum), Team Coach or Project Lead

Acts as a coach for guiding the team and obtaining resources whenever required.

The role often demands the soft skills of project management more than planning and technical skills, which are often left to the team.

The person is not necessarily the team’s manager but rather the role reflects knowledge and responsibility over rank.

**Team Member**

Responsible for the project’s creation and delivery.

The team members will normally be comprised of developers, QA, and documentation.

They are responsible for planning, design, development, testing, and on-time project delivery.

Product Owner (Scrum), On-Site Customer, Active Stakeholders

Represents the voice of the customer and is responsible for the prioritized backlog and maximizing the return on investment (ROI).

Part of this role’s responsibility includes documenting user stories and/or requirements for the project.

**Stakeholders**

Represents a broad category of people who can be users, managers of users, operations, support, Portfolio Managers, other agile teams with dependencies, executive team, investors, and more.

Agile teams will sometimes have extended cast members who are called upon to provide technical or domain expertise for certain specialized skills that may not be present among the team members.

Likewise, it is not always reasonable or fair to expect product owners to be so-called experts in every aspect of a product or domain. This is when they call in domain experts to assist the team with certain requirements.

Agile teams will find that different approaches are available and specific solutions work best for them.

But the impact of Agile on product development cannot be understated, with its focus primarily being the customer and the art of collaboration within the team.

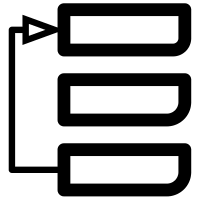
**Agile Principles**

* Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
* Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
* Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
* Business people and developers must work together daily throughout the project.
* Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
* The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
* Working software is the primary measure of progress.
* Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
* Continuous attention to technical excellence and good design enhances agility.
* Simplicity the art of maximizing the amount of work not done is essential.
* The best architectures, requirements, and designs emerge from self-organizing teams.
* At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.”

**Agile Artefacts**

Agile scrum artefacts are information that a scrum team and stakeholders use to detail the product being developed, actions to produce it, and the actions performed during the project. They provide important data which give insights of the performance of the team. They enable core scrum attributes of transparency, inspection, and adaption.

Main Artefacts of scrum are,

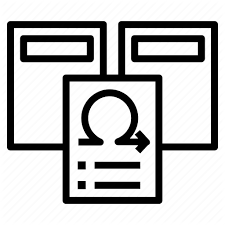


**Product Backlog**

The [product backlog](https://www.atlassian.com/agile/scrum/backlogs) is a list of new features, enhancements, bug fixes, tasks, or work requirements needed to build a product. It’s compiled from input sources like customer support, competitor analysis, market demands, and general business analysis. t contains tasks that were once in an active sprint but deprioritized and moved to the backlog.

* It should be prioritised
* It should be overestimated
* It should be high level and dynamic.

**Sprint Backlog**

****

The sprint backlog is a set of product backlog tasks that have been promoted to be developed during the next product increment. Sprint backlogs are created by the development teams to plan deliverable for future increments and detail the work required to create the increment.  Sprint backlogs are created by selecting a task from the product backlog and breaking that task into smaller, actionable sprint items.

* Some best practices while creating a sprint backlog can be:
* Make group decision
* Don't Assign Task
* Definition of done
* Keep updating the backlog
* Add any task



**Burn-down Chart**

A burn down chart is a graphic representation of how quickly the team is working through a customer’s user stories, an [agile](https://www.projectmanager.com/ultimate-agile-guide) tool that is used to capture a description of a feature from an end-user perspective.

**Agile Development Tools**

In agile development, the emphasis **on** building the right product as per customer needs. Therefore, agile testers need to monitor their project constantly. There are many tools available for this purpose. Good agile tools organize the discussion and planning. The developers can focus on each of the features, tasks, or bugs in separate threads. Splitting the discussions up helps the project move forward at the right rate for each section.

**Source control tools**

Git, like some of the other tools here, wasn't built just for agile teams but is still essential. It offers much of the flexibility that teams need to move ahead. The lack of one dominant central repository makes it simpler for different developers to follow different paths and then merge their code later.

Other top source control tools include Mercurial, Subversion, and CVS.

**Continuous integration tools**

Just like Git, continuous integration tools aren't explicitly designed to support agile development, but it would be hard to imagine running a large agile team without their help. The tools automatically add a layer of processing when code is committed, helping to ensure that the team is working smoothly together. There are a number of good continuous integration tools that play well with agile management systems. Some of the best-known tools include Hudson, Jenkins, Travis CI, Strider, and Integrity.

How do Agile Project Management tools help?

Agile development tools help you in building the right product as per customer needs. Agile testers can use these tools to monitor their projects constantly.

Which factors to consider while selecting the Agile Project Management Tool?

Don't slog to choose the right tool. Get smart. Consider the following factors before you decide.

Agile Project Management tool should allow for team management.

* Software rate.
* Software features meet your requirements.
* It should be able to generate basic reports like bug reports and activity reports.
* Tool Vendor support and update policy.
* The cost involved in training employees on software.

What are the important Agile tools and techniques?

Important Agile tools are Zephyr, TestRail, Kanbanize, Backlog, JIRA, and Soap UI.

Important Agile techniques are follows:

* Assess project progress, demonstrations, and retrospectives on a regular basis
* Arrange training sessions to ensure the business team has the skills needed to manage Agile activities.
* Break down your requirements into clear tasks that will make it much easier to evaluate the effort needed to complete the work.
* Create a blueprint of the project before your team starts working on it.

**Team Activity 4 Agile and V-Model Tools**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tool** | **Use** | **Description** | **Reference** |
| **Agile Tools** | | | |
| Gantt Chart | 1. Sprint Tracking | Gantt charts are a simple, visual way to track tasks across the lifecycle of a project. In Agile PM, teams are doing sprints, which are basically short clusters of related tasks. A Gantt chart is a helpful way to capture and plan this work. | https://technologyadvice.com/blog/information-technology/agile-teams-use-gantt-chart-software/#:~:text=Gantt%20charts%20are%20a%20simple,capture%20and%20plan%20this%20work.&text=You%20can%20also%20track%20the%20change%20requests%20you%20receive%20from%20project%20stakeholders. |
| 2. Gantt Chart as a Collaboration Tool | After you build this Gantt chart to track sprints, you can also use it to plan and organize work with your team. Some Gantt charts are in and of themselves a powerful collaboration tool.  An online Gantt chart can serve as a platform where the team plans together and tracks each other’s progress throughout the sprint, noting deliverables at the task level, attaching files with comments, notes, etc. |
| 3. Integrated Timesheets | Gantt charts as part of a project management software tool can be integrated with timesheets to help monitor the time your team is putting into the work. You can then determine if they are over-tasked or if the project needs more resources. |
| Burndown Charts | To see how qucikly team burns through stroies | The burndown is a chart that shows how quickly you and your team are burning through your customer's user stories. It shows the total effort against the amount of work we deliver each iteration. Something like this: | [http://www.agilenutshell.com/burndown](https://www.mercurial-scm.org/about) |
| Mercurial | source control management | Mercurial is a free, distributed source control management tool. It offers you the power to efficiently handle projects of any size while using an intuitive interface. It is easy to use and hard to break, making it ideal for anyone working with versioned files. |
| AGILE MANAGER | will push stories and tasks directly to these tools so developers can keep track directly from their favorite IDE. | During each code sprint, the scrum masters and developers record their progress on the user stories and issues. All the progress (or failures) from the build and the unit tests are plotted in charts on a dashboard so the entire team can watch how they're converging on the release. |
| **V Model Tools** | | | |
| Microsoft Project, Word | Project Planning | Define the problem and scope of the existing system. Overview the new system and determine its objectives. Confirm project feasibility and produce the project Schedule.. | https://www.tutorialspoint.com/system\_analysis\_and\_design/system\_analysis\_and\_design\_development\_life\_cycle.htm |
| Microsoft Project is a standard tool for maintaining the project plan. Microsoft Project is a project management software product, developed and sold by Microsoft. It is designed to assist a project manager in developing a schedule, assigning resources to tasks, tracking progress, managing the budget, and analyzing workloads. |
| Can also keep in word documents. |
| Rational Rose, Eclipse UML, TogetherJ | Requirement Analysis | Gather, analyze, and validate the information. Define the requirements and prototypes for the new system. Evaluate the alternatives and prioritize the requirements. Examine the information needs of the end-user and enhance the system goal. | <https://www.perfecto.io/blog/right-testing-tool-each-phase-sdlc> |
| Rational Rose XDE, an "eXtended Development Environment" for software developers, integrates with Microsoft Visual Studio .NET and Rational Application Developer. The Rational Software division of IBM, which previously produced Rational Rose, wrote this software. |
| UML2 Tools is a Graphical Modeling Framework editor for manipulating UML models. |
| Together is a product from Micro Focus, formerly from Borland (acquired by Micro Focus in 2009), formerly from TogetherSoft (acquired by Borland in 2003), that currently integrates a Java IDE, which originally had its roots in JBuilder,[citation needed] with a UML modeling tool. |
| Visio, Rational Rose, Eclipse UML, TogetherJ | Design | Includes the design of application, network, databases, user interfaces, and system interfaces. Transform the SRS document into logical structure, which contains detailed and complete set of specifications that can be implemented in a programming language. Create a contingency, training, maintenance, and operation plan. | <https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_development_life_cycle.htm> |
| Visio is used to draw a flowchart. Microsoft Visio is a diagramming and vector graphics application and is part of the Microsoft Office family. |
| Eclipse, Visual studio, JDeveloper, Net Bean | Implementation | Code is written based on the system and design.Combine all the modules together into training environment that detects errors and defects. | <https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_development_life_cycle.htm> |
| Eclipse is an integrated development environment used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. |
| Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. |  |
| Oracle JDeveloper is a free integrated development environment that simplifies the development of Java-based applications addressing every step of the application lifecycle. JDeveloper offers complete end-to-end development for Oracle's platform and Oracle's applications. |
| NetBeans is an integrated development environment for Java. NetBeans allows applications to be developed from a set of modular software components called modules. |  |
| CVS, VSS, Clear Case, PVCS | Source code repository and version control systems | Manage the source code from multiple developers. | <https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_development_life_cycle.htm> |
| Concurrent Versions System (CVS) is a program that lets a code developer save and retrieve different development versions of source code. It also lets a team of developers share control of different versions of files in a common repository of files. This kind of program is sometimes known as a version control system . |
| Microsoft Visual SourceSafe (VSS) is a discontinued source control program, oriented towards small software development projects. Like most source control systems, SourceSafe creates a virtual library of computer files. |  |
| ClearCase is a software configuration management tool used for version control. It manages changes across development lifecycles, from design to code to test. It is one of many version control systems available today. |
| PVCS (formerly Serena PVCS Pro) is a change and configuration management solution designed for small development teams. PVCS integrates software version control, issue tracking, and defect management into one easy-to-use tool. |  |
| Nunit, NunitAsp, Microsoft fxcop, jetbrains dotTrace, SSW  code Auditor. | Testing | A test report which contains errors is prepared through test plan that includes test related tasks such as test case generation, testing criteria, and resource allocation for testing. | <https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_development_life_cycle.htm> |
| Nunit – unit testing tool for all .net language. |
| NunitAsp – automatically testing the asp web pages. |  |
| Microsoft fxcop – Code analysis for .net managed code. |
| SSW code Auditor – Automatically review the web application and projects. |  |
| Jetbrains dotTrace profiler – Improving the performance of .net application. |

**Individual Activity5**

**Cyclometric Complexity**

* Cyclometric complexity is a source code complexity measurement that is being correlated to a number of coding errors. It is calculated by developing a Control Flow Graph of the code that measures the number of linearly-independent paths through a program module.
* Lower the Program's cyclometric complexity, lower the risk to modify and easier to understand

Code :

switch (action) {

case 0: { exit(0); }

case 1:

{

addNewMember(store);

break;

}

case 2:

{

addNewBook(store);

break;

}

case 3:

{

borrowBooktoaMember(store);

break;

}

default:

break; }

{

this\_thread::sleep\_for(chrono::seconds(4));

}

Cyclometric Complexity Diagram:

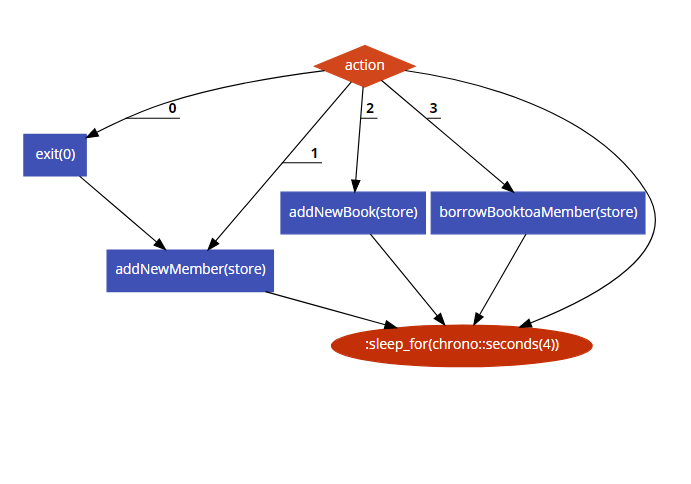


Fig 14. Cyclometric Complexity

**Calculation:**

Cyclometric complexity = E - N + 2\*P

where,

  E = number of edges in the flow graph.

  N = number of nodes in the flow graph.

  P = number of nodes that have exit points

Cyclometric complexity = E - N + 2\*P

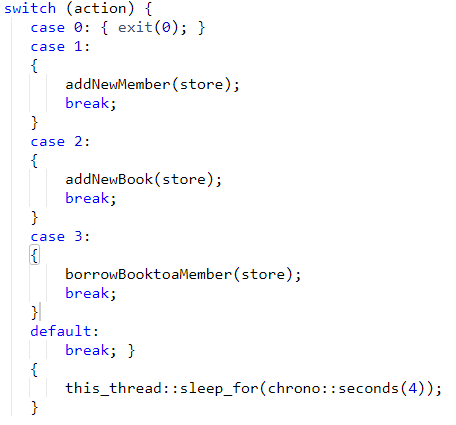
= 9-6+(2\*1)

= 3+2

= 5

**MC/DC**

Multiple Condition Decision Coverage (MCDC) is also known as Modified Condition Decision Coverage. In MCDC each condition should be evaluated at least once which affects the decision outcome independently.



In the above code, there is a switch condition, to satisfy condition coverage each statement in the above code must be TRUE at least one time.

Test case for the above code is

**TEST CASE1** = Action = 0;

**TEST CASE2** = Action = 1;

**TEST CASE3** = Action = 2;

**TEST CASE4** = Action = 3;

**TEST CASE5** = Action = default;

To satisfy the decision coverage we need to ensure that the all the switch case statement evaluates to TRUE and FALSE at least once. So, in the above test cases you can see that the change in the value of Action variables made a change in decision outcomes.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| User Action | Case 0 | Case1 | Case2 | Case3 | default | Result |
| Action=0 | Action=0 | True | False | Action=0 | Action=0 | True = **TEST CASE1** |
| Action=1 | Action=1 | Action=1 | Action=1 | Action=1 | Action=1 | True = **TEST CASE2** |
| Action=2 | Action=2 | Action=2 | Action=2 | Action=2 | Action=2 | True = **TEST CASE3** |
| Action=3 | Action=3 | Action=3 | Action=3 | Action=3 | Action=3 | True = **TEST CASE4** |
| Action! = input list | Action! = input list | Action! = input list | Action! = input list | Action! = input list | Action! = input list | True = **TEST CASE5** |

**Team Activity5 Calculator**

Calculators are a widely used device nowadays. It makes calculations easier and faster. Calculators are used to everyone in daily life. A simple calculator is made using a C++ program which can add, subtract, multiply and divide, two operands entered by the user. The switch and break statement is used to create a calculator.

**Research:**

**Aging and Costing:**

Human civilization invented the abacus, which the Computer History Museum suggests is “the oldest continuously used calculating tool aside from fingers.”

## **1623: First Adding Machine**

## **1773: First Functional Calculator**

## **1820: First Commercially Produced Mechanical Calculator**

## **1954: First All-Transistor Calculator**

## **1961: First All-Electronic Desktop Calculator**

## **1967: First Handheld Calculator**

## **1971: First Truly Pocket-Sized Electronic Calculator to use LED Display**

## **1974: First Handheld Programmable Calculator**

## **1985: First Graphing Calculator**

## **2003: First Graphing Calculator with Touch Functionality**

## **2010: First Color Graphing Calculator**

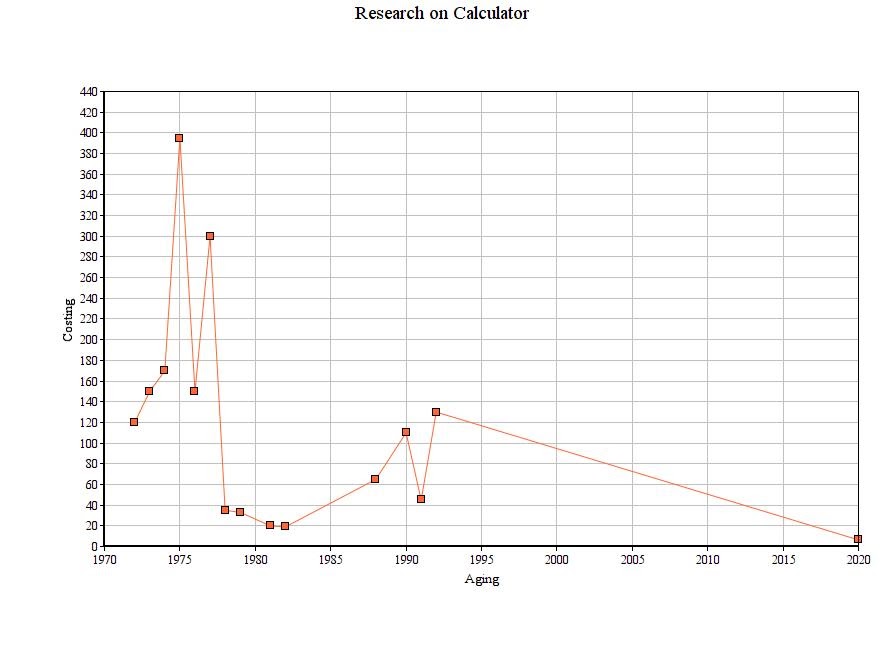


Fig 1. Ageing and Cost

References:

<https://www.thocp.net/hardware/ti_calculators.htm>

<https://edtechmagazine.com/k12/article/2012/11/calculating-firsts-visual-history-calculators>

**Specification:**

* 8-Digit display, 8-Digit math accuracy
* Simple left to right math calculations
* Fixed or floating point operation
* 4 Basic operands Addition, Subtraction, Multiplication, Division
* 3 Scientific functions Sin ,Tan ,Cos
* Result
* button input
* Displays math expression
* Clear error messages
* Savable Settings

**SWOT Analysis:**

S

* Decreased Complexity of Solving Problem
* Rising Usage of Mobile Phone
* Growing Dependency in Banking, Accounting, and Financial Institutes
* High Adoption of Digital Calculators
* Increased Demand of Calculator Due To Digitization

W

* Competitive segment

O

* Can be used for Personal, School or Business use
* The Growth in the Personal Business leads to the High Demand of Calculators
* Rising Demand of Solar Calculator

T

* Competitive segment
* Increasing Automation in the Various Instruments

<https://www.advancemarketanalytics.com/reports/45087-global-calculators-market>

# Requirements:

# High Level Requirements:

|  |  |
| --- | --- |
| **ID** | **Description** |
| H1 | Display given input by user on Screen |
| H2 | Able to select keys for getting output and displaying the output on the screen. |
| H3 | Able to select the two different calculators i.e. Standard and Scientific                 Calculator |
| H4 | Able select the different signs of operation on keyboard of Calculator |
| H5 | Displaying the Math expressions and Showing the error messages when any error |

**Low Level Requirement:**

|  |  |
| --- | --- |
| **ID** | **Description** |
| L1 | Able to add two numbers |
| L2 | Able to subtract two numbers |
| L3 | Able to Multiply two numbers |
| L4 | Able to Divide two numbers |
| L5 | Able to find sine of given angle |
| L6 | Able to find cosine of given angle |
| L7 | Able to find tangent of given angle |
| L8 | Display error and result |

# Design

**High Level Design Diagram**

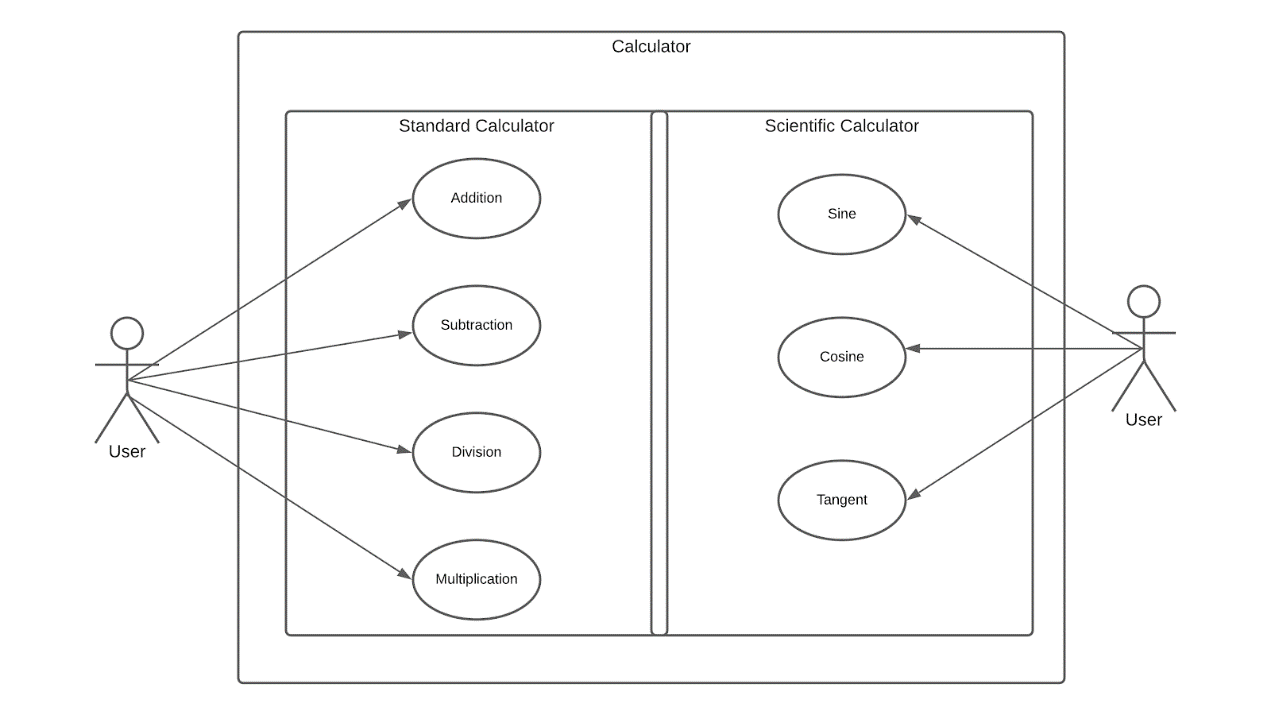


Fig 2. Use Case UML Diagram

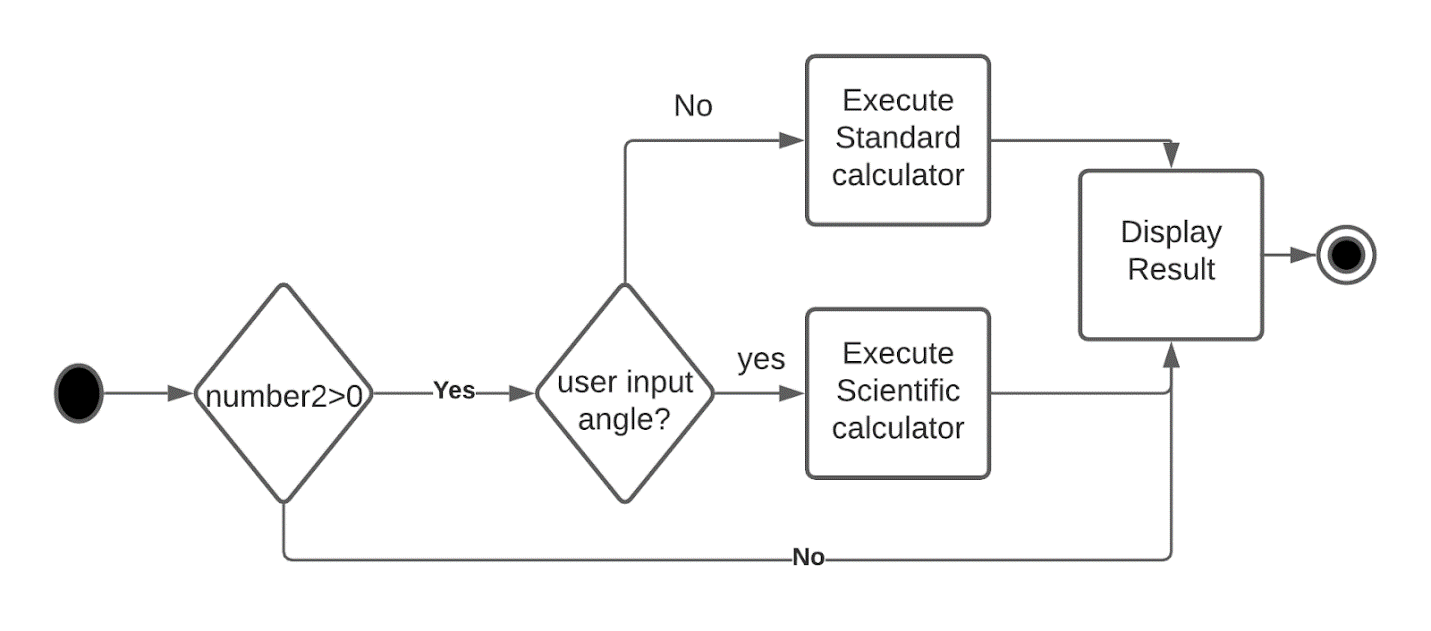


Fig 3. Activity UML Diagram

**Low Level Design UML Diagram**

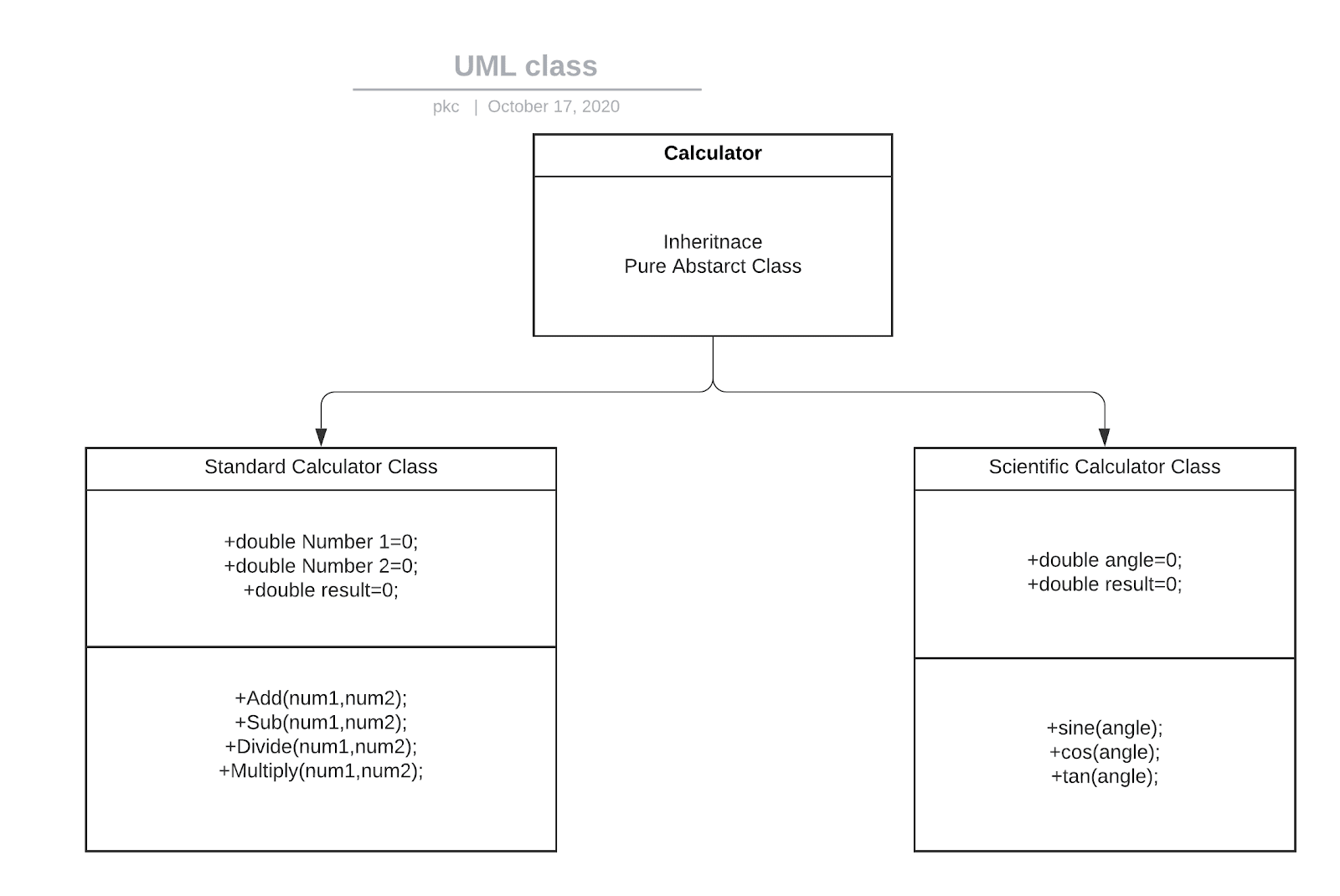


Fig 4. Class UML Diagram

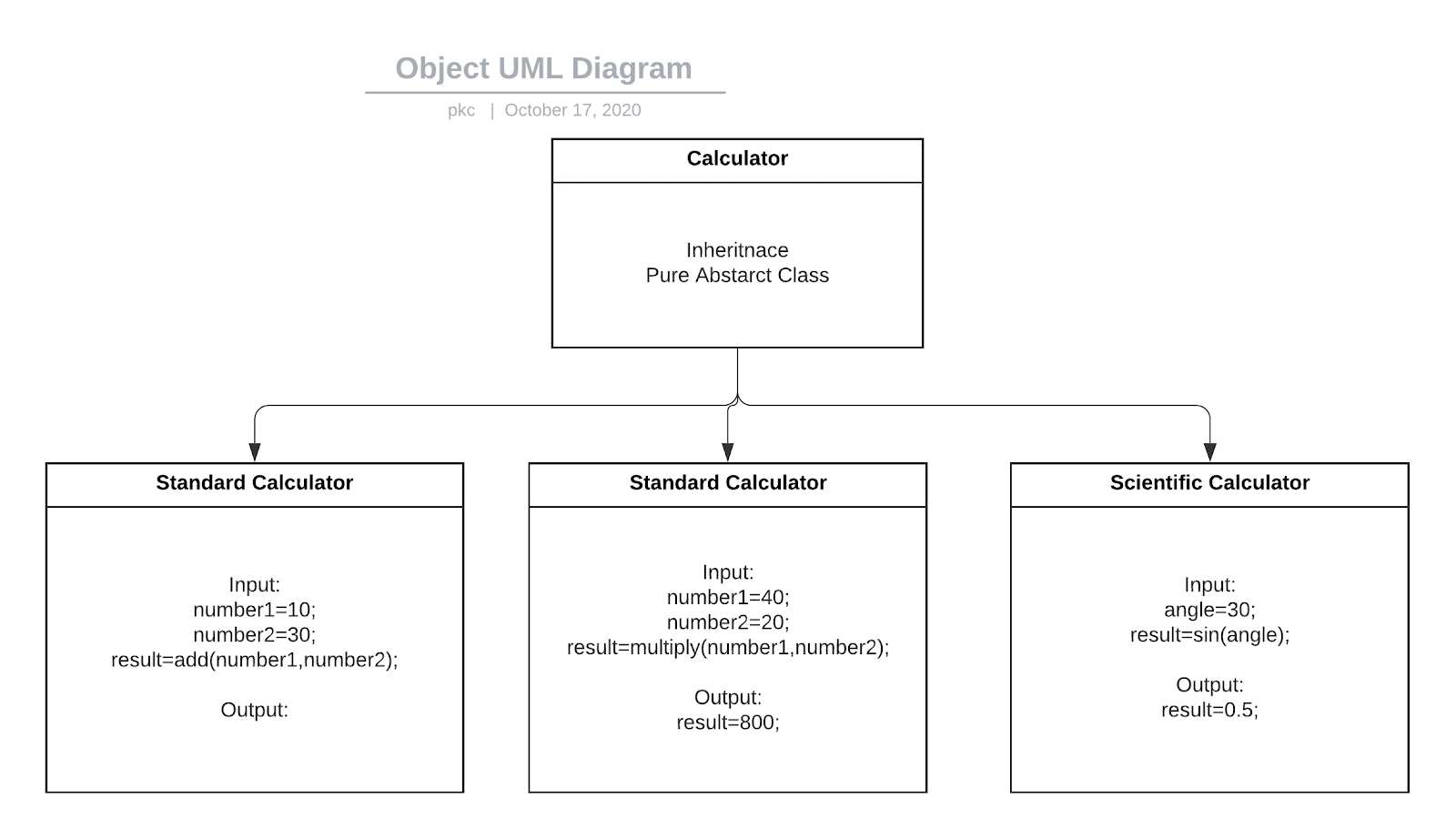


Fig 5. Class Object Diagram

# Test Plan:

# Unit Test Plan:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Precondition** | **Expected IO** | **Expected OP** | **Actual OP** |
| **U1** | To **Add** two integers and display the result on the output screen calculator. | The Calculator program should be compiled successfully without errors. | 135 + 100 | 235(addition of above ten digits will be expressed in exponential form) |  |
| **U2** | To **Subtract** two integers and display the result on the output screen calculator. | The Calculator program should be compiled successfully without errors. | 135 - 100 | 35 |  |
| **U3** | To **Multiply** integers and display the result on the output screen calculator. | The Calculator program should be compiled successfully without errors. | 135 \* 100 | 13500 |  |
| **U4** | To **Divide** integers and display the result on the output screen calculator. | The Calculator program should be compiled successfully without errors. | 135 / 100 | 1.35 |  |
| **U5** | Scientific Operation  Find the **sine** value of an angle provided | The scientific class should be compiled and without errors. | Sin (30) | 0.5 |  |
| **U6** | Scientific Operation  Find the **cosine** value of an angle provided | The scientific class should be compiled and without errors. | Cos (30) | 0.8660 |  |
| **U7** | Scientific Operation  Find the **tangent** value of an angle provided | The scientific class should be compiled and without errors. | Tan (30) | 0.5773 |  |

# Integration Test Plan:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Precondition** | **Expected IO** | **Expected OP** | **Actual OP** |
| **I1** | Able to compute the given functions and display the output on the screen | Calculator program should be compiled with zero errors. | Inputs are given in the Unit Test file. | Passing Successfully. |  |
| **I2** | Check the calculator class when no parameters are passed. | Calculator program should be compiled with zero errors. | Add (no parameters) | Error: The function add with zero parameters not available. |  |
| **I3** | Passing the function with other parameters like char or string. | Calculator program should be compiled with zero errors. | Sub (char, char) | Error: The function sub with illegal parameters not allowed. |  |
| **I4** | Computation Test on Divide by zero | Calculator program should be compiled with zero errors. | 135/0 | Undefined. |  |
| **I5** | Computation Test for Scientific Calculator | The scientific class should be compiled and without errors | Tan (multiple parameter) | Not allowed. |  |

**Agile methodology:**

**THEME**:  
 Calculator application with Two functionalities standard and scientific.

In standard calculator functions are basics calculations like addition,subtraction ,multiplication and division.

In scientific calculator functions are Sin ,Tan and Cos.

Epic:

· 8-Digit display, 8-Digit math accuracy

· Simple left to right math calculations

· Fixed or floating-point operation

· 4 Basic operands

   Addition, Subtraction, Multiplication, Division

· 3 Scientific functions

   Sin, Tan, Cos

· Result

· button input

· Displays math expression

Clear error messages

· Savable Settings

**USER STORIES:**

|  |  |  |  |
| --- | --- | --- | --- |
| USER TYPE | EPIC | USER STORY |  |
| Accountant | Calculate the interest and  profits | Through he wants to see the result with less time | |  |  | | --- | --- | | STORY | STORY POINT | | Calculating profits | 3 | | Calculating Interest | 2 | |
| Primary school teacher | Teacher want to teach students about basic addition, subtraction, multiplication and division | Given 2 number 16 and 8.  When I add, multiply, subtract and divide.  Then the result should be 24,144,8,2 respectively in each output. | |  |  | | --- | --- | | STORY | STORY POINT | | When applying addition and subtraction | 3 | | When doing multiplication and division | 3 | |
| Students | Use for scientific calculation | Students want to calculate trigonometry functions like Sin, Tan, Cos. | |  |  | | --- | --- | | STORY | STORY POINT | | Applying Sin | 3 | | Applying Cos and Tan | 2 | |

Product Backlog

|  |  |  |
| --- | --- | --- |
| Priority | User Story | Story Points |
| 1 | When a User tries any of the functions which include +, %  , x ,- ,sin ,cos ,tan the user can expect the proper functionalities . | 4 |
| 2 | The user inputs two numbers (e.g.: 50 and 10) separated by the ‘+’ operator, the user should expect an output of 60 in this case. | +2 |
| 2 | The user inputs two numbers (e.g.: 50 and 10) separated by the ‘-’ operator, the user should expect an output of 40 in this case. | +2 |
| 2 | The user inputs two numbers (e.g.: 50 and 10) separated by the ‘%’ operator, the user should expect an output of 5 in this case. | +2 |
| 2 | The user inputs two numbers (e.g.: 50 and 10) separated by the ‘x’ operator, the user should expect an output of 500 in this case. | +2 |
| 2 | The user inputs ‘sin’ function followed by 90, and should expect an output of 1. | +2 |
| 2 | The user inputs ‘cos’ function followed by 90, and should expect an output of 1. | +2 |
| 2 | The user inputs ‘tan’ function followed by 90, and should expect an error output. | +2 |
| 3 | The user presses the ‘AC’ button hence clears the contents on the screen. | +1 |
| 3 | The user presses the ‘=’ button after entering arguments and respective functions and should expect the output to be displayed. | +1 |

**Sprint Planning**

**Stakeholder involved:** Product Owner, Scrum master, Scrum team.

**Sprint Backlog: Sprint 1: 3** EPICS ---> To do: 5 Completed: 0

**Sprint 2: 2** EPICS ---> To do: 4 Completed: 0

**Sprint Reviews**: Sprints need to be completed.

**Sprint Burndown Chart:**  2 x 7 + 1 x 2 + 4 = 20 story points left out.

**Story Points: 20**

**Reference:**

* + - [***https://techbeacon.com/app-dev-testing/top-agile-tools-keep-software-engineers-productive***](https://techbeacon.com/app-dev-testing/top-agile-tools-keep-software-engineers-productive)
    - [***https://www.guru99.com/agile-project-management-tools.html***](https://www.guru99.com/agile-project-management-tools.html)
* [***http://www.kscst.iisc.ernet.in/spp/41\_series/40S\_awarded\_&\_selected\_projs\_further\_devpt/40S\_BE\_0889.pdf***](http://www.kscst.iisc.ernet.in/spp/41_series/40S_awarded_&_selected_projs_further_devpt/40S_BE_0889.pdf)
* [***https://www.researchgate.net/publication/319535629\_Automated\_Smart\_Water\_Vending\_Machine***](https://www.researchgate.net/publication/319535629_Automated_Smart_Water_Vending_Machine)
* [***https://www.quora.com/What-is-the-difference-between-UML-and-SysML***](https://www.quora.com/What-is-the-difference-between-UML-and-SysML)
* [***https://www.academia.edu/30476478/REAL\_TIME\_EMBEDDED\_BASED\_DRINKING\_WATER\_VENDING\_MACHINE***](https://www.academia.edu/30476478/REAL_TIME_EMBEDDED_BASED_DRINKING_WATER_VENDING_MACHINE)