Learning Report
Applied SDLC with Software Testing

GLOBAL ENGINEERING ACADEMY

Genesis

L&T Technology Services

Name	PS Number	Email ID
Asha N	99002646	asha.n@ltts.com

Document History

Ver. Rel. No.	Release Date	Prepared. By	Reviewed By	To be approved By	Remarks/Revision Details
1	15/10/2020	Asha N	Akash Shetty and Chappidi Lokanath Reddy	Dr Prithvi Sekhar	
2	16/10/2020	Asha N	Akash Shetty and Chappidi Lokanath Reddy	Dr Prithvi Sekhar	
3		Asha N			

Table of Contents

TEAM ACTIVITY - 2 : POWER OF VISUAL REPRESENTATION

TEAM ACTIVITY - 3 : AGILE METHODOLOGY

TEAM ACTIVITY - 4 : SIMPLE CALCULATOR

TEAM ACTIVITY - 5 : TOOLS

REFERENCES :

1. Introduction

Beverage Dispenser is a vending machine that gives beverages like coffee, milk, tea, hot water, cream, sugar to users after cash is inserted into the machine. Some of the machines, notably older models, utilize powdered instant coffee mixed with hot water, and a few of those provide condiments like cream and sugar. Some newer models fresh-brew the coffee using hot water and ground coffee beans, and some also grind the coffee to order using coffee grinders installed in the machines, as well as providing various condiments. Some modern machines also provide other hot drinks such as tea, espresso, lattes, cappuccinos, mochas and hot chocolate. Some of the machines dispense canned coffee, and some dispense both hot coffee and iced coffee.

2. Research

2.1 Ageing

The first vending machine in the U.S. was built in 1888 by the Thomas Adams Gum Company, The first modern coin-operated vending machines were introduced in London, England in the early 1880s, dispensing postcards.

Vending machines exist in many countries and, in more recent times, specialized vending machines that provide less common products compared to traditional vending machine items have been created.

2.2 Costing

Coin Operated Vending Machine (Floor standing vending machine)

Type:	Food and Drinks
Environmental Condition:	Semi-outdoor
Payment:	Coin, Bill, Cashless Payment
Charge System:	Coin and Note
Function:	Insulation, Heating
Touch Screen:	Touch Screen
Specification:	1920x720x620mm
Power-off Protection:	With Power-off Protection
Powder Canister:	3 KG/ Canister
No. Of Canister:	1 Bean Canister+4 Powder Canisters
Temperature Control System:	Hot drinks 105 ºC max
Water Supplying:	Pump

Price: US $2,500-3,000

2. Tabletop Coffee-Tea Vendor

LCD Display:	14 inch touch screen
Power-off Protection:	With Power-off Protection
Maximum Power:	2700W
Standard cup size:	10oz(80mm diameter cup size)
Grinder:	Ditting
Coffee brewer:	Jetinno patent
Daily maintenance:	Automatically clearance
Espresso drink speed:	45s
Instant Drinks Speed:	25S (120ml)
Cup capacity:	120cups per day
Price:	US $880.00-$890.00

3. Definition of the Product:

- Adding sensors to to measure beverages.
- Adding New features like Cold drinks and also Accepting exact amount of beverage.
- Implementing feature to select the amount distribution.
- Upgrading output method for different drinks

4. SWOT Analysis

Strength	**Weaknesses**
● Basic need of consumer ● Different types of flavors ● Low cost ● Non-Alcoholic	● Age of life cycle ● Time taken to register ● Depend on power supply
Opportunities	**Threat**
● Replace alcoholic drinks ● Growing possibilities if cold drinks ● Upgrading output method for different drinks	● Competition with barista , Mochas , Gloria, Jean , Costa Coffee. ● Presence of other 'Hangout' locations ● Competition with Starbucks , Lavazza, Caribou coffee, Dinkin Donuts etc..

Table 2: SWOT Analysis

5. Requirements

5.1 High level requirements:

ID	Description
HL_01	Different types of cold drinks.
HL_02	Different Output method for different drinks.
HL_03	Display the amount to pay.
HL_04	Display the selection panel to select amount distribution.

Table 3: High Level Requirements

5.2 Low level requirements:

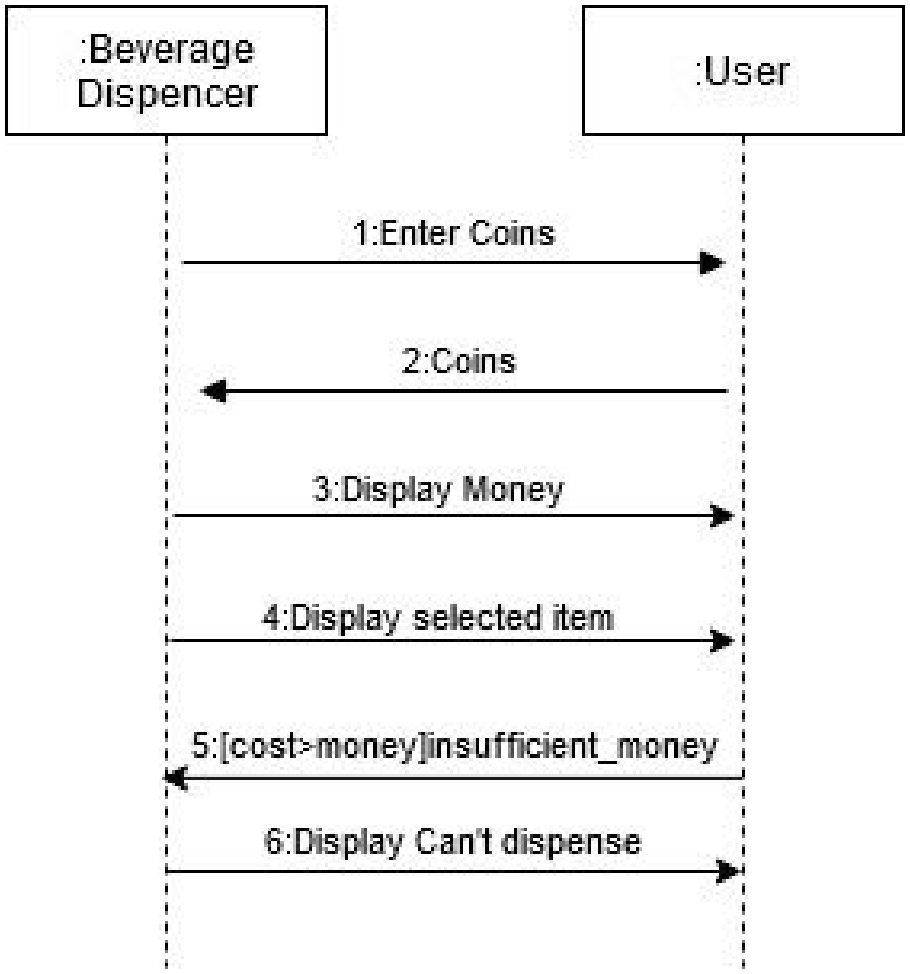
ID	Description
LL_01	Sensor to measure beverages.
LL_02	Software maintenance.
LL_03	Depending on power.

Table 4: Low Level Requirements

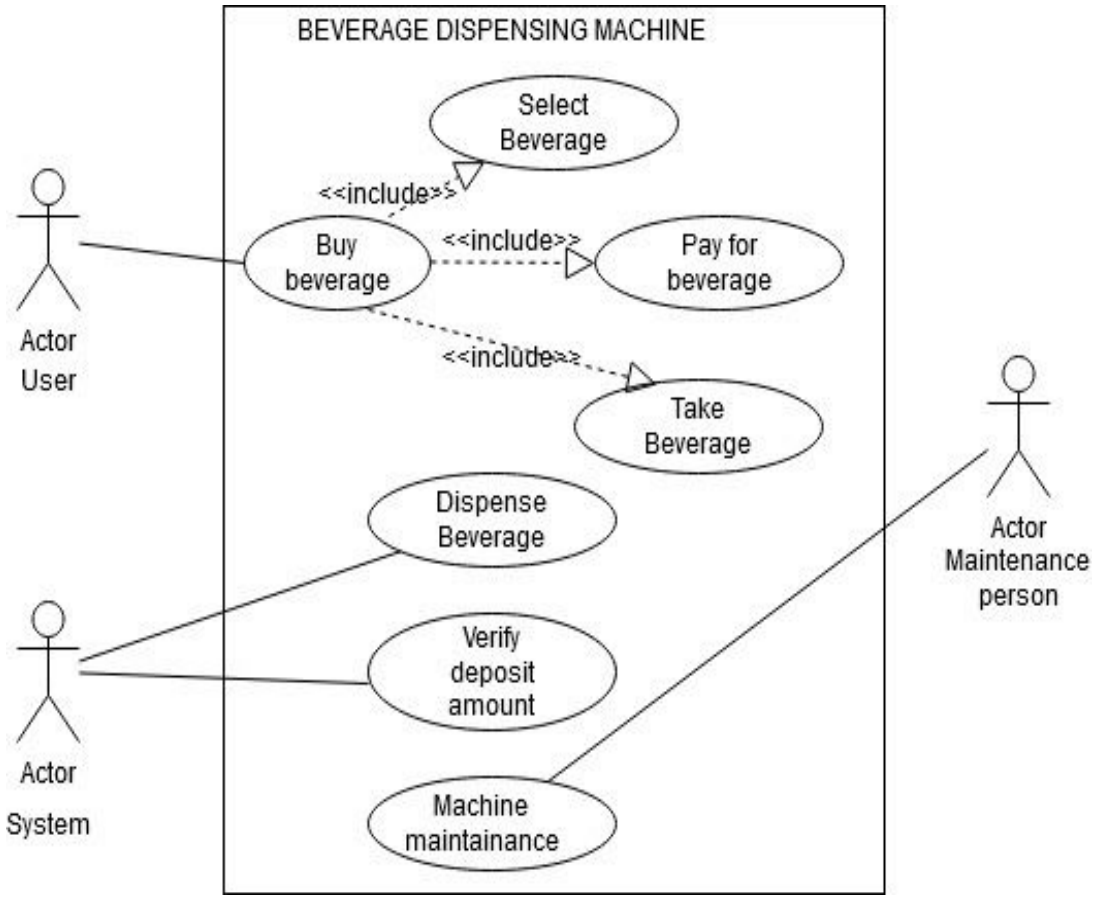
6. UNIFIED MODELING LANGUAGE DIAGRAM

6.1 Behavioral diagram

1. Sequence diagram

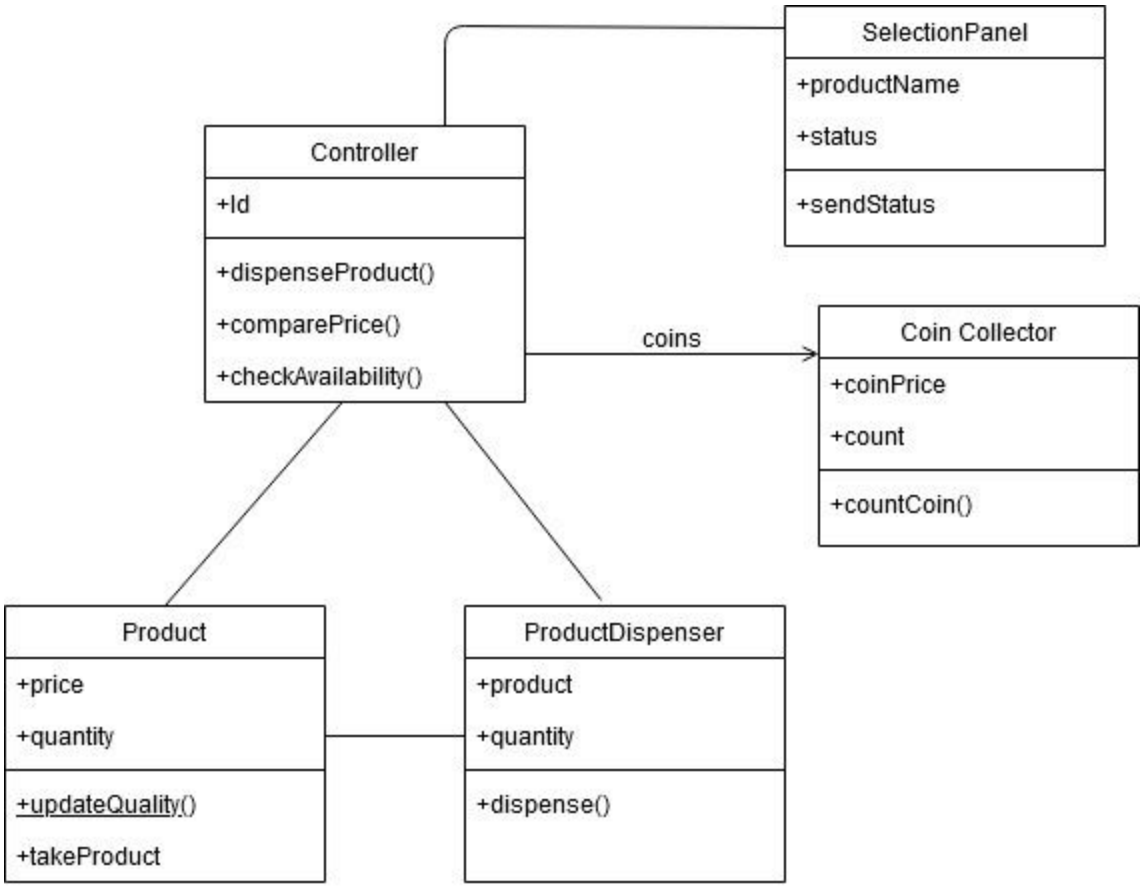


2. Use case Diagram

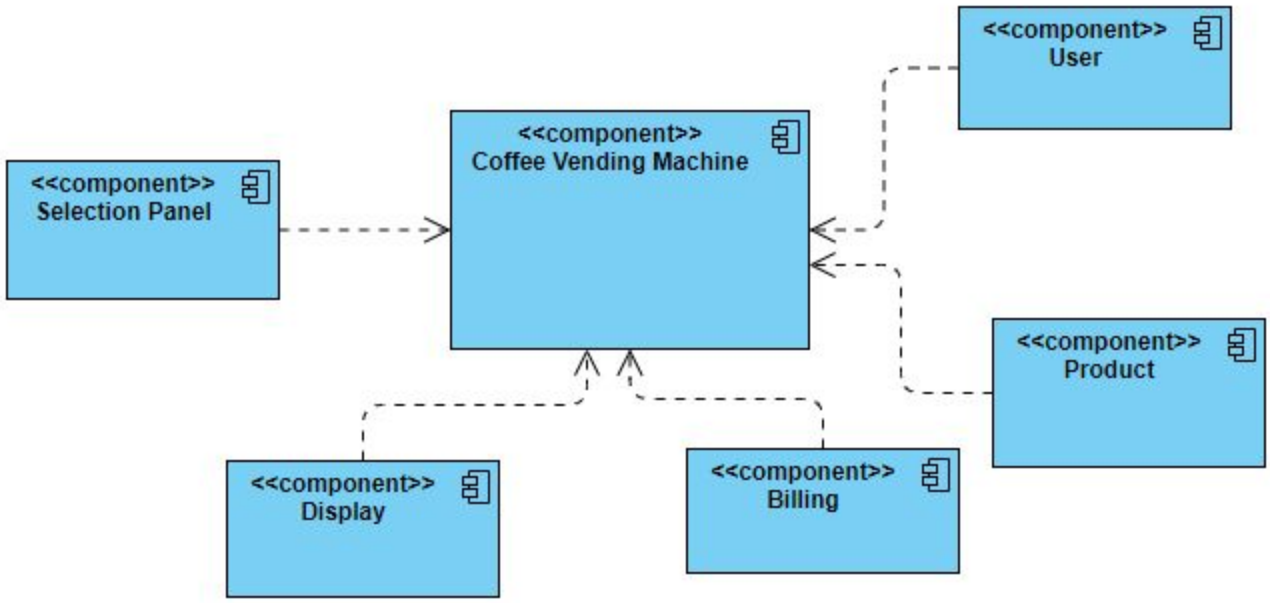


6.2 Structural diagram

1. **Class diagram**



2. Component diagram



7. Test Plan

7.1 Unit testing:

Test id	Description	Expected input	Expected output	Actual output
HH_01	On selection of cool drinks	different types of cool drinks	Display of different types of cool drinks	Display of different types of cool drinks

HH_02	Different output for cool drinks	Choosing between different types of cool drinks	Cool drinks should come from different path	Cool drinks should come from different path
HH_03	Display the amount to pay	Selection of beverage	Displays the correct amount	Displays the correct amount
HH_04	On display of money selection	Clicking on Menu	Display of different types of money distribution	Display of different types of money distribution

Table 5: Unit Testing

7.2 Integration testing:

Test id	**Description**	**Expected input**	**Expected output**	**Actual output**
LL_01	Sensor Implementation	To add Cup first	True	True
LL_02	On Software Maintenance	Maintenance person login	Display of quantity of cool drinks and coffee	Display of quantity of cool drinks and coffee
LL_03	Depends on Power	If power cut Immediately	Restart the machine	Restart the machine

Table 6: Integration testing

Team Activity1 - Major defects and recalls in Aerospace

Impact of defective Product

1. Incidents

Boeing Recall – Dreamliner (2016): Failure of the engine.

Boeing Recall - Max 737 (2019): Faulty sensor gives erroneous data about positions.

Airbus Recall - A320neo, 2018: The unavailability of parts for repairs.

2. Causes

Erroneous data.

Glitches in the design.

Failure of maintenance.

3. Impacts

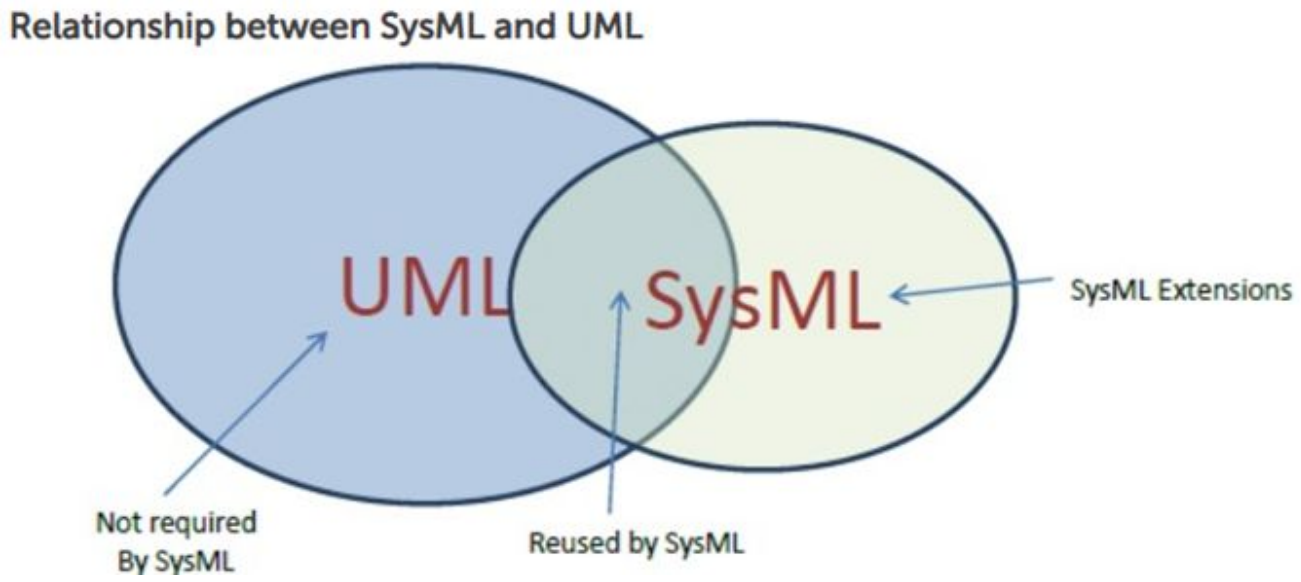
Recalls of more than 200.

357 casualties combined.

Market valuation decreases.

Team Activity 2: Power of visual representation

The Systems Modelling Language (SysML) is a general purpose modelling language for engineering systems. SysML supports the analysis, design and verification of complex systems including hardware, software, information, personnel, procedures, and facilities in a graphical notation.SysML is defined as an extension of a subset of the Unified Modelling Language (UML) using UML's profile mechanism.



- SysML is a comparatively small language that is easier to learn and apply. Since SysML removes many of UML's software-centric constructs, the overall language is smaller both in diagram types and total constructs.
- SysML allocation tables support common kinds of allocations. Whereas UML provides only limited support for tabular notations, SysML furnishes flexible allocation tables that support requirements allocation, functional allocation, and structural allocation. This capability facilitates automated verification and validation (V&V) and gap analysis.
- SysML model management constructs support models, views, and viewpoints. These constructs extend UML's capabilities and are architecturally aligned with IEEE-Std-1471-2000 (IEEE Recommended Practice for Architectural Description of Software Intensive Systems)
- SysML reuses seven of UML 2's fourteen diagrams, and adds two diagrams (requirement and parametric diagrams) for a total of nine diagram types. SysML also supports allocation tables, a tabular format that can be dynamically derived from SysML allocation relationships.

ACTIVITY-1 V PROCESS CHAPTER-1

INTRODUCTION

The Hospital Management System is focused on the principle of making appointments for patients. Here, the user can update doctors after logging in as an administrator. Other features include visualising the hospital's complete doctor data and making / attending appointments.

1.1 Definition

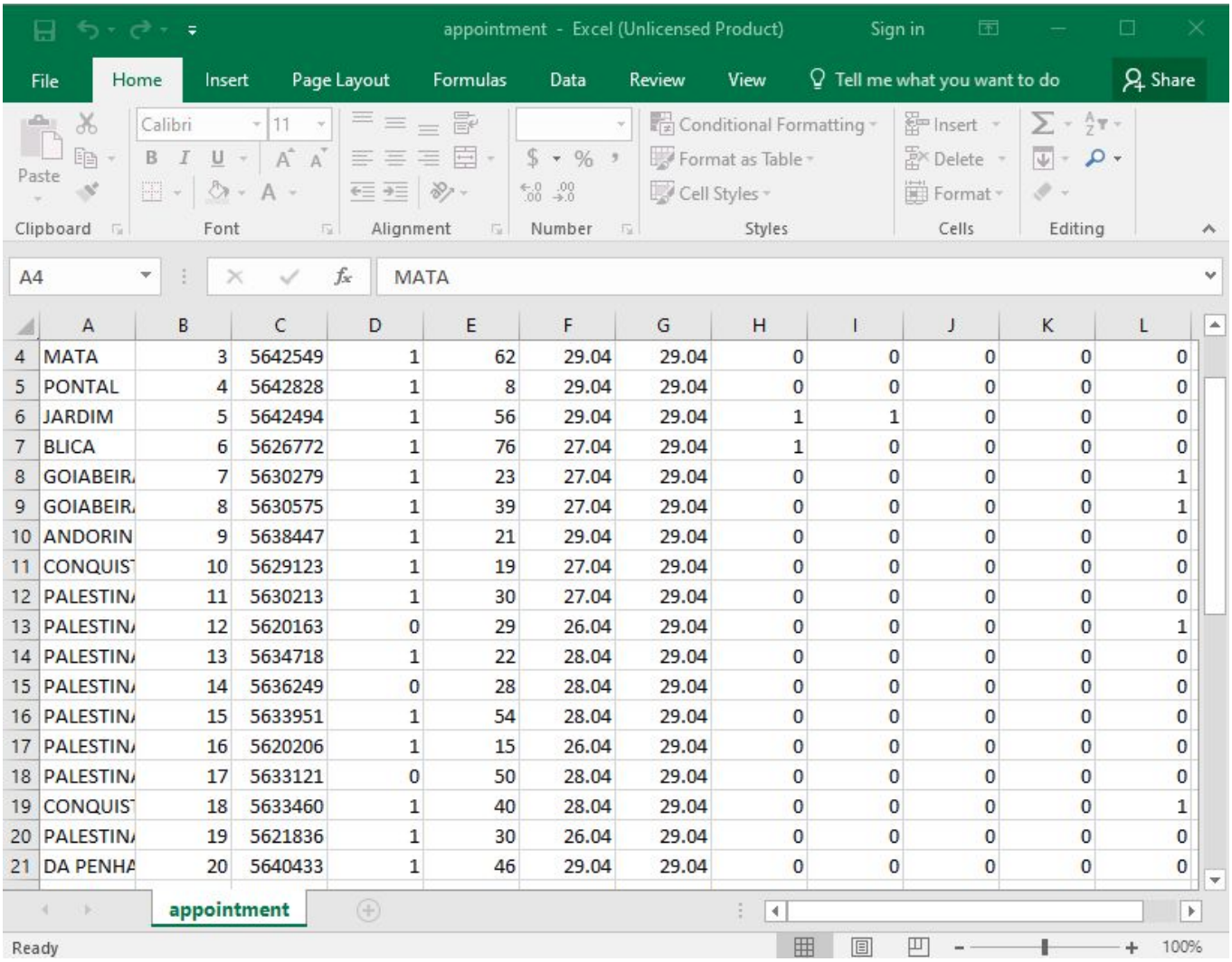
A user can view the complete data of the doctor, including ID number , name and appointment time. Having an appointment and attending it is the key aspect of this project. Before they move on it. The user must provide his / her name, choose a gender and provide a number. He/she can make an appointment quickly, but he/she needs to pick a doctor and enter the doctor's ID for appointments. Similarly, he / she must enter the number of the doctor when attending an appointment.

Features:

1. Login System
2. Make/Attend Appointment
3. Update Doctor
4. View all doctor’s information

CHAPTER-2
REQUIREMENTS

Input to the Doctor appointment is taken from a csv file where all the details stored in a specified format.



	A	B	C	D	E	F	G	H	I	J	K	L
4	MATA	3	5642549	1	62	29.04	29.04	0	0	0	0	0
5	PONTAL	4	5642828	1	8	29.04	29.04	0	0	0	0	0
6	JARDIM	5	5642494	1	56	29.04	29.04	1	1	0	0	0
7	BLICA	6	5626772	1	76	27.04	29.04	1	0	0	0	0
8	GOIABEIR	7	5630279	1	23	27.04	29.04	0	0	0	0	1
9	GOIABEIR	8	5630575	1	39	27.04	29.04	0	0	0	0	1
10	ANDORIN	9	5638447	1	21	29.04	29.04	0	0	0	0	0
11	CONQUIS	10	5629123	1	19	27.04	29.04	0	0	0	0	0
12	PALESTIN	11	5630213	1	30	27.04	29.04	0	0	0	0	0
13	PALESTIN	12	5620163	0	29	26.04	29.04	0	0	0	0	1
14	PALESTIN	13	5634718	1	22	28.04	29.04	0	0	0	0	0
15	PALESTIN	14	5636249	0	28	28.04	29.04	0	0	0	0	0
16	PALESTIN	15	5633951	1	54	28.04	29.04	0	0	0	0	0
17	PALESTIN	16	5620206	1	15	26.04	29.04	0	0	0	0	0
18	PALESTIN	17	5633121	0	50	28.04	29.04	0	0	0	0	0
19	CONQUIS	18	5633460	1	40	28.04	29.04	0	0	0	0	1
20	PALESTIN	19	5621836	1	30	26.04	29.04	0	0	0	0	0
21	DA PENHA	20	5640433	1	46	29.04	29.04	0	0	0	0	0

CHAPTER-3

VERIFICATION AND VALIDATION

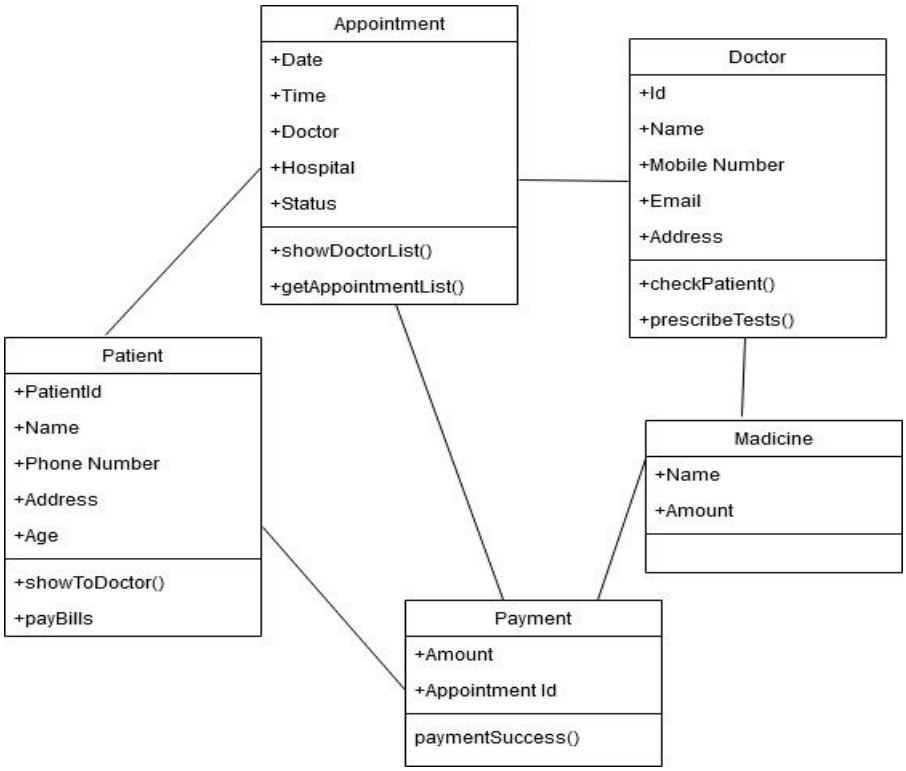


Figure : Class Diagram

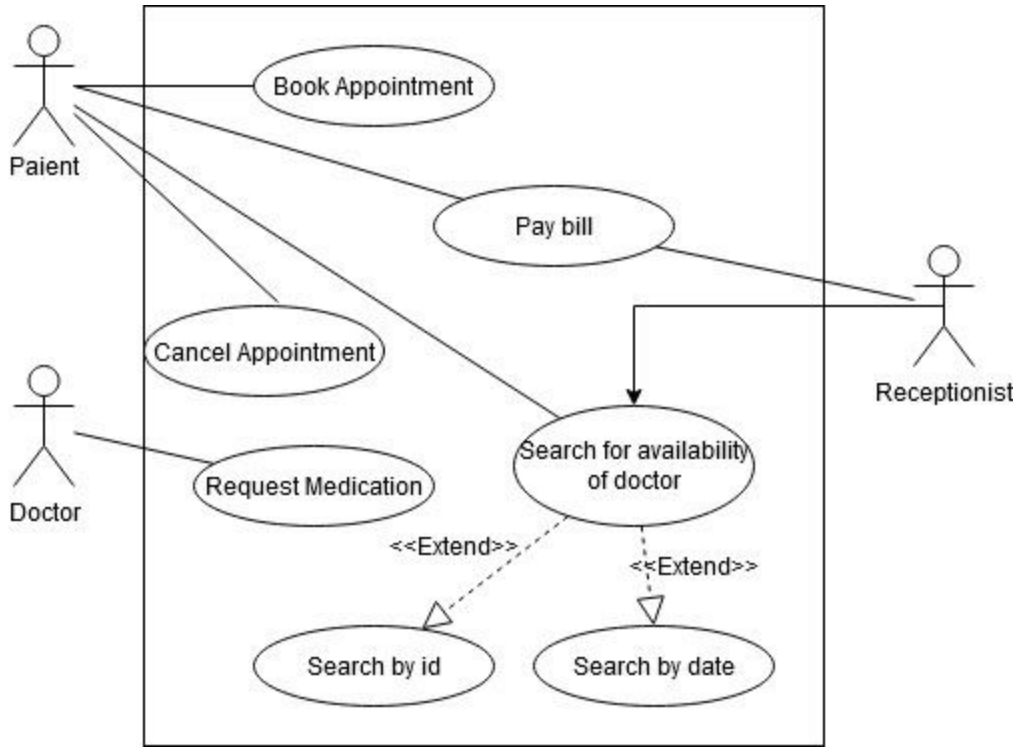


Figure :Use case diagram

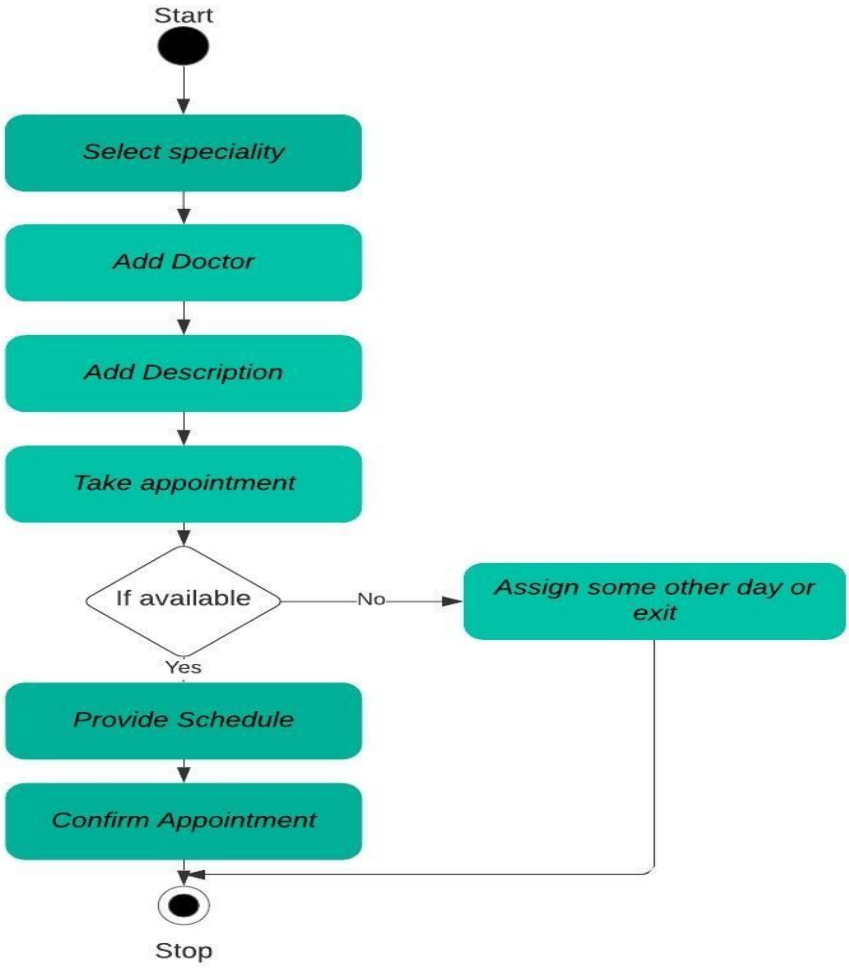


Figure : Activity diagram

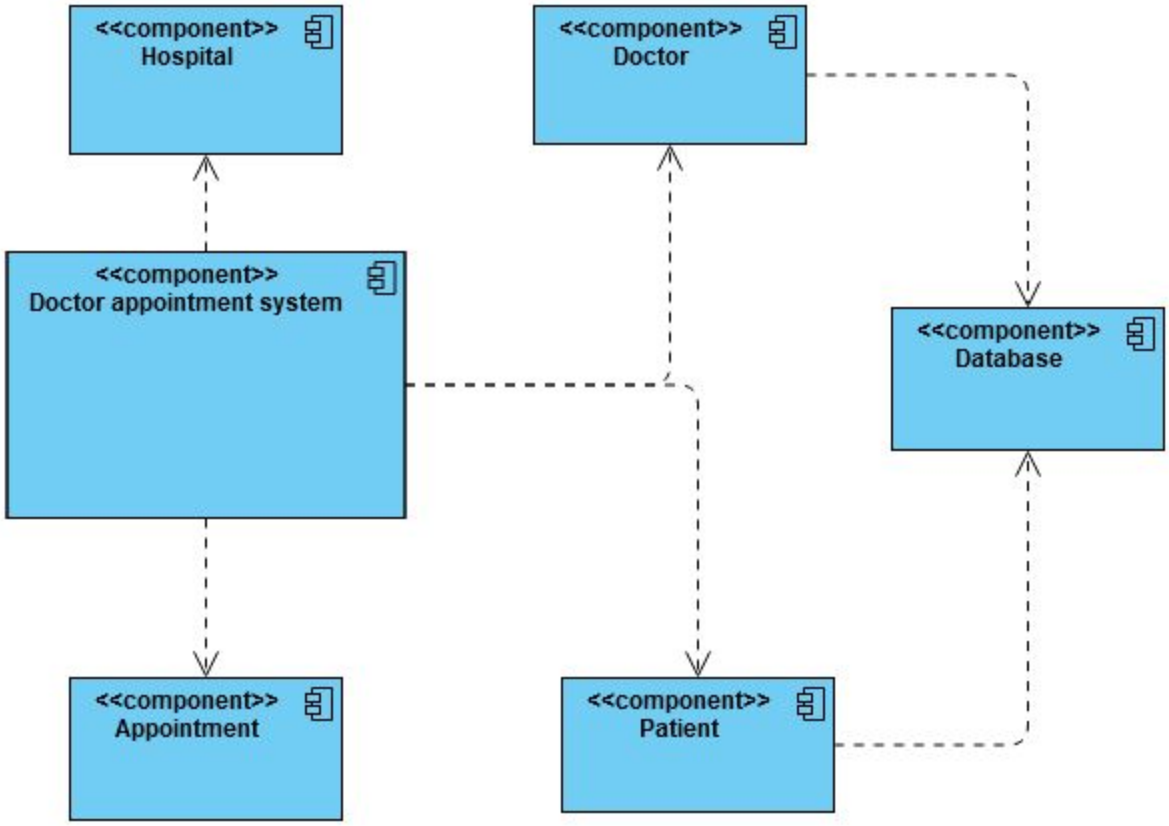
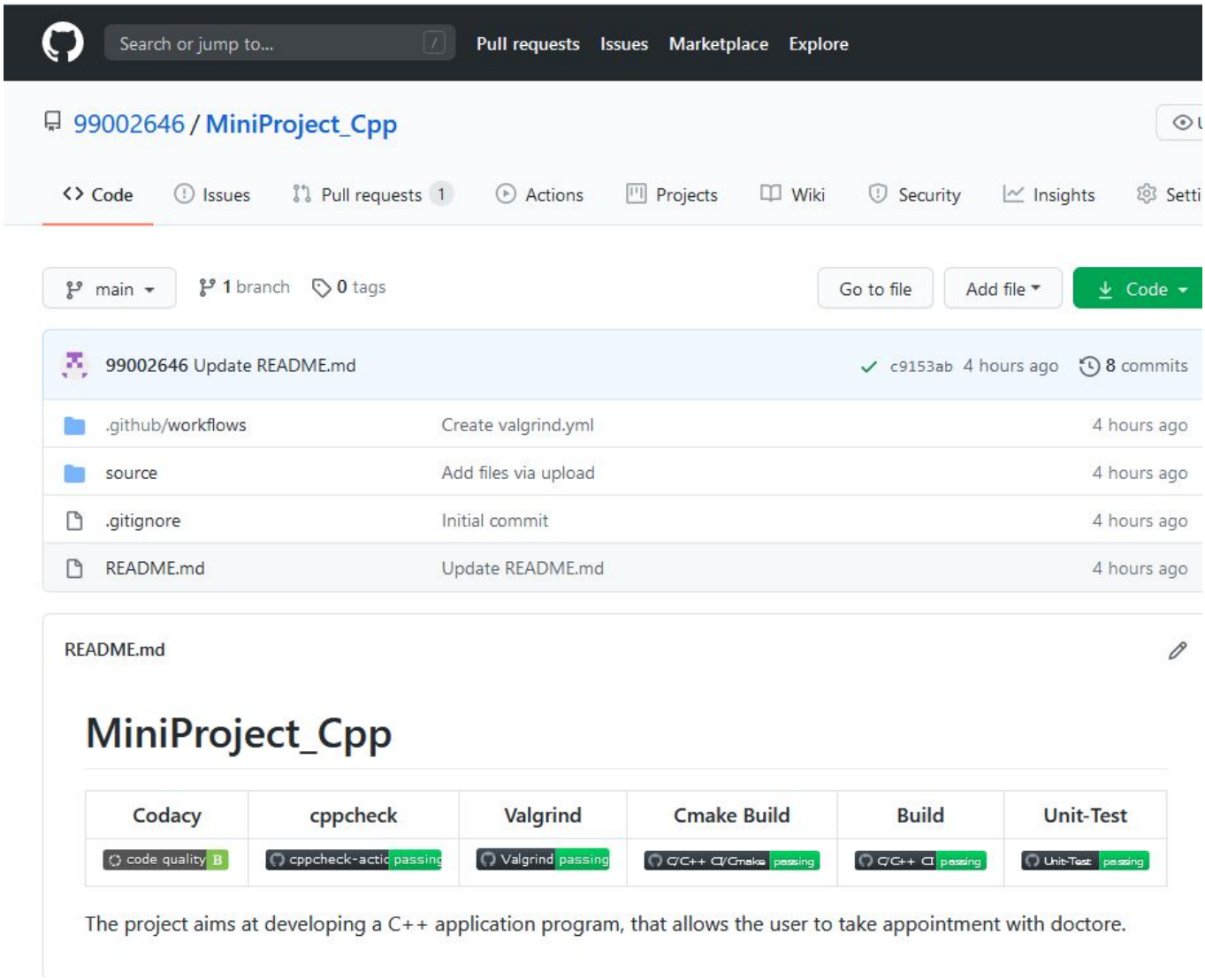


Figure : Component Diagram

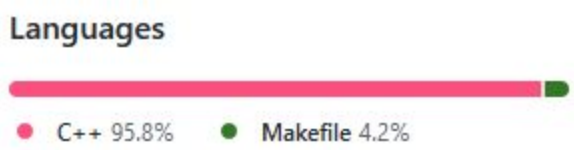
CHAPTER-4

GIT ASPECTS

GitHub repo link : https://github.com/99002646/MiniProject_Cpp



Doctor appointment system github dashboard with badges



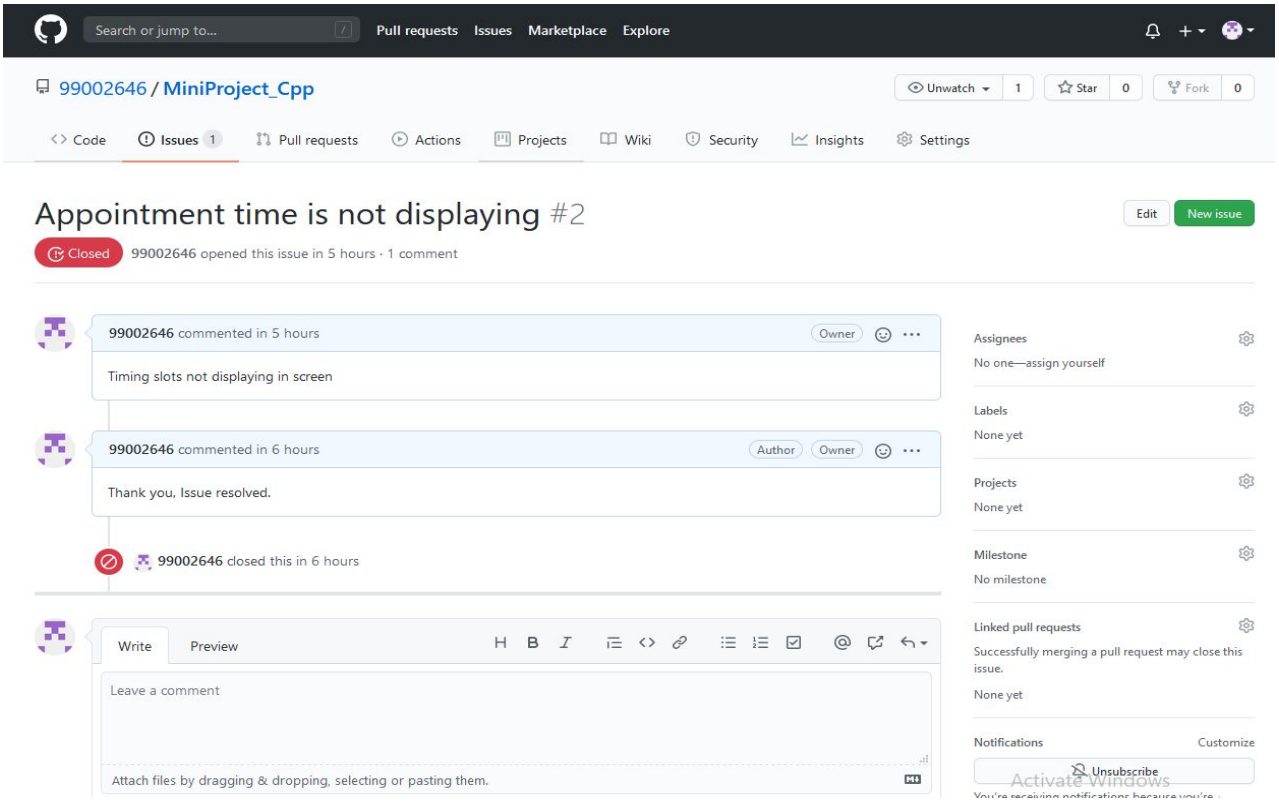


Figure : Issue snapshot

CHAPTER-5

CONCLUSION

The Doctor Appointment System is moderned and improved system, so anyone from any place can book the appointment and meet the doctor. It is very useful.

ACTIVITY-2 : APPLYING V-MODEL ON C++ PROJECT

About Product:

- **Checking the what is yield, production on respected area and year**
- **Analysis of different types of crops yield based on year.**
- **Analysis of particular crop on area, production and yield.**

High level requirements:

ID	Description
HL_01	Analysis of 5 year's production of particular crop.
HL_02	Comparision of different crops from 5 years data
HL_03	Highest and lowest production,area and yield of particular crop.
HL_04	Adding of new crops details as if needed.

Low level requirements:

Low level requirements:

ID	Description
LL_01	Reading data from csv file.
LL_02	Saving all data on list using STL concepts

LL_03	**Implementation of CI/CD.**

Table 4: Low Level Requirements

System Design:

1.Structural UML

Class diagrams

Component diagrams:

Behavioral Diagrams

1.Sequence Diagrams

2.Use case diagrams

Testing

Unit testing:

Test id	Description	Expected input	Expected output	Actual output
HH_01	**Knowing of data of crop**	**Adding the data to list**	**Display of list where crop is added**	**Crop added**

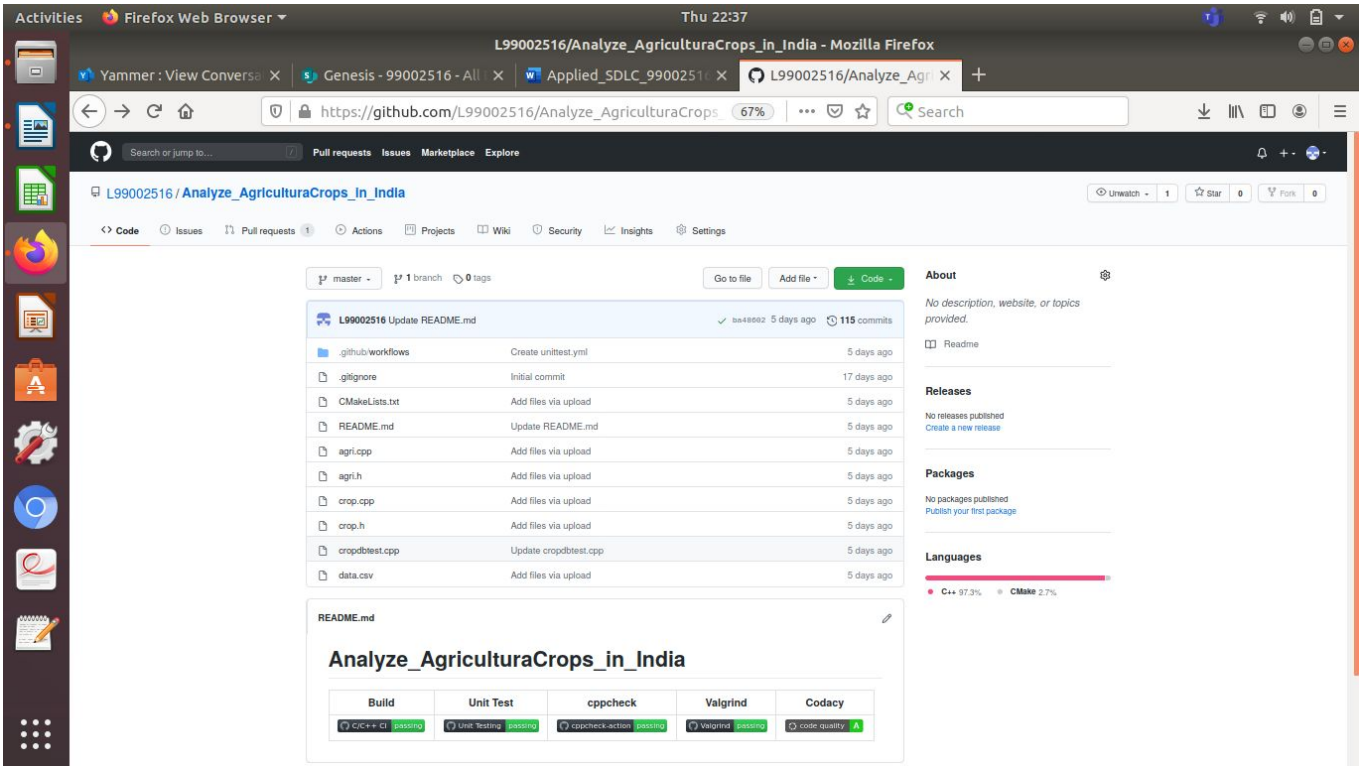
HH_02	**Analysis of different crop**	**Checking of different crops**	**Present of different crops**	**True**
HH_03	**Highest production,area and yield**	**Giving crop name**	**Giving highest year where production or area or yield .**	**Year dispaly**
HH_04	**Adding of new crop**	**Adding of new crop**	**Display of list where new crop is added**	**True**

Table 5: Unit Testing

Integration testing:

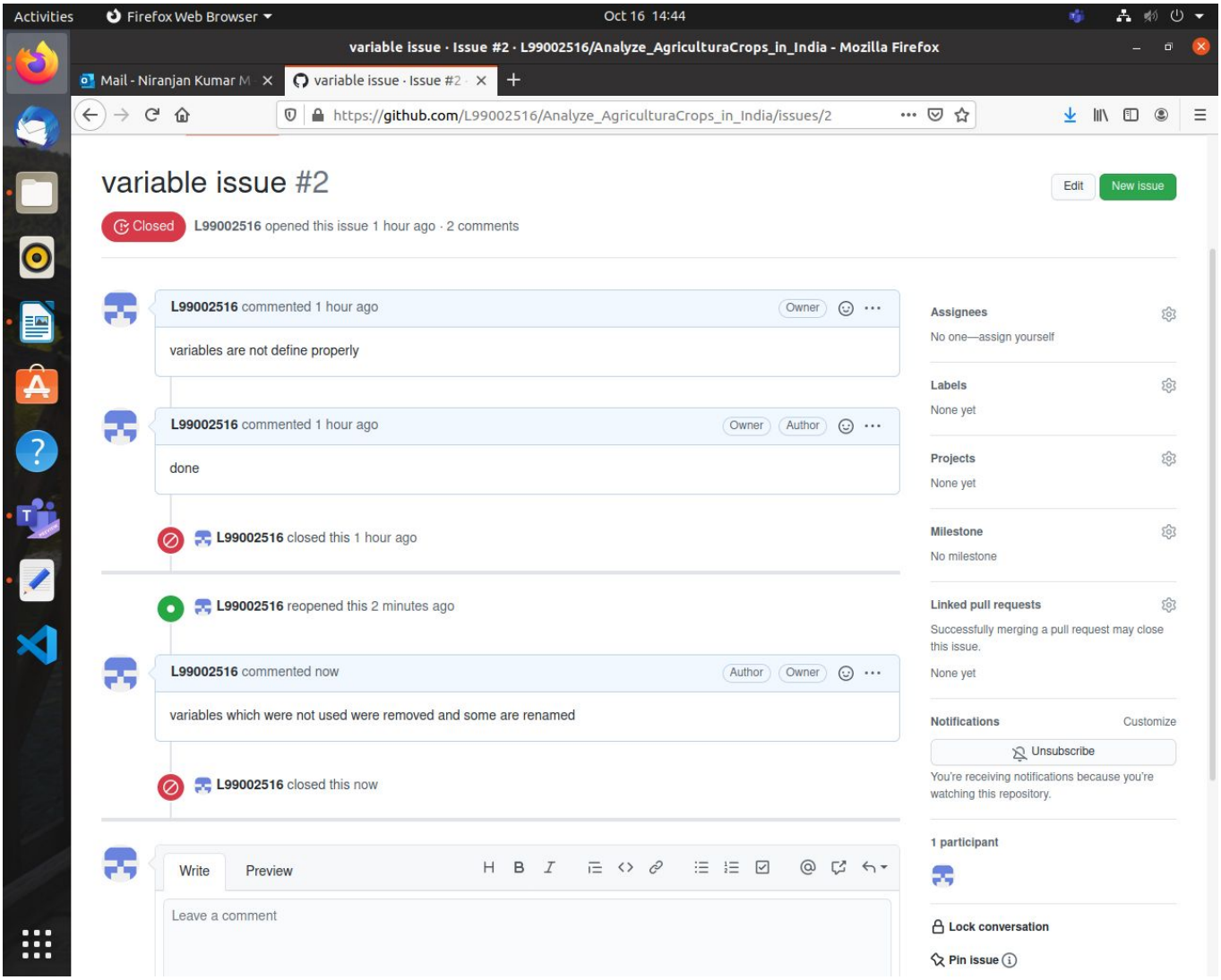
Test id	Description	Expected input	Expected output	Actual output
LL_01	**Reading of csv file**	**Csv file**	**Adding of all data present in csv to list**	**Data added to list**
LL_02	**Adding data to list using STL concepts**	**Adding data to list**	**Data added to list**	**Display of list**
LL_03	**CI/CD**	**GitHub Actions**	**Cppcheck, valgrind, unit testing , codacy**	**Passing all CI/CD**

Continuous Integration/ Continuous delivery



GitHub:https://github.com/L99002516/Analyze_AgriculturaCrops_in_India.git

C++ project.



Activity-3 Agile

Theme

The beverage dispenser uses the grinder to grind the coffee beans so the coffee can be prepared fresh. With that the beverage dispenser dispenses the different types of cool drinks and also uses the different path to dispense different beverages.

Epic

- This begins when the customer wants to purchase drinks.
- The customer selects the drink.
- Then the dispenser shows the quantity of the drink.
- The customer selects the quantity and the next task will proceed.
- Dispenser checks for the availability of the drinks and shows the error if the condition is not satisfied.
- If the condition is satisfied then the dispenser displays the amount to pay.
- The customer should pay the correct amount.
- If the paid amount is lesser than the bill amount, then the error message will be displayed.
- If the customer pay the correct amount, then the drink which is selected by the customer is produced.
- Then the completion message will be displayed on the display.
- Then the machine completes the use case.

User Story and sprints

Sprint-1 Amount not enough

- If the paid amount is lesser than the bill amount, then the error message will be displayed.
- After the dispenser cancels the transaction.

Sprint-2 Drinks not in stock

- The customer selects the drinks he wants.
- The dispenser checks for the availability of drinks.
- The dispenser shows the error message.

Sprint-3 Power cut

- When the power cuts, the dispenser will shut.
- After the power connection comes back then the dispenser will start from the beginning.
- Then the path will be cleaned.

Activity-3(Team) Agile for 'Aroma'

Firstly, Agile software development, also known as Agile, is an outlook to software development, one that unfolds requirements and solutions through the collaborated effort of self-organising, cross-functional teams and their clients or end users. It recommends planning using adaptive methods along with evolutionary development, empirical knowledge, and continual progress.

What is Agile Manifesto?

The Manifesto for Agile Software Development, commonly referred to as Agile Manifesto, Is a legal official order that includes twelve principles and four values to show the way for an iterative and people-centric approach to software development. It focuses primarily on testing while keeping the code simple, delivering the functioning bits of the application as soon as they are ready. It promotes an easy, clear and simple approach to developing software in short sprints so that each functioning bit of the software could be analysed and tested based on the client's or the end user's requirements, and may be changed if required to meet their needs.Although this set of values and principles were formed primarily for software development, the same can be applied to different forms of business.

This makes Agile a very effective and flexible method for all forms of business.

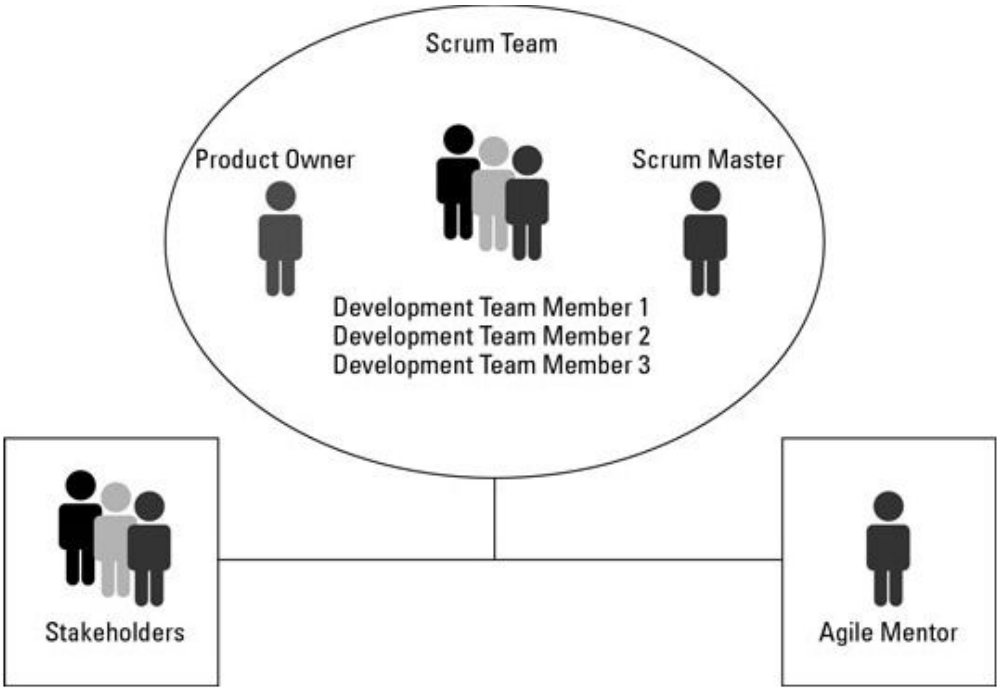
Agile Manifesto Values

What is Agile Principle?

The following 12 Principles are based on the Agile Mainfesto.

1. highest priority: **satisfy the customer**
2. even **late change of requirements** is welcomed
3. Frequent **delivery of** working **software**
4. daily **work together**
5. motivated individuals is given environment and support they need, and trust them to get the job done
6. conveying information: **face-to-face conversation**
7. primary measure of progress: working software
8. agile processes promote **sustainable development**, stakeholders should be able to maintain a constant pace **indefinitely**
9. continuous attention to technical excellence and good design enhances agility.
10. simplicity - the art of maximising the amount of work not done - is essential
11. **self-organising teams** → best architectures, requirements, and designs
12. team regularly reflects on how to become more effective

What is Agile Roles?



Scrum Master:

- The Scrum Master is considered to be the top-dog in every organization because companies usually hire them and don't treat them as permanent employ that is why they are with no authority.
- It is their duty to remove all the hindrance or obstruction in the way of achieving any goal.
- It is also their role to enforce scrum ceremonies and processes.
- They are the ones who commit to goals and deadlines on behalf of the team.

Product Owner:

- The product owner is responsible for conveying the vision of the stakeholders to the team.
- They have the authority to alter the scope.
- The Product Owners are responsible for the return on investment (ROI) that is why they occupy an authoritative position in the firm.
- Because they convey the vision of the stakeholders that is why they are the voice of the stakeholders.
- Not only with the team, but they also communicate with the stakeholders about progress and problems.

Scrum Team:

- The Scrum Team is responsible for all the activities that lead them towards their sprint goals.
- They have to work with the Scrum Master to prioritize the items from the product backlog in the sprint planning.
- Once committed, it is their responsibility to fulfill the commitment and deliver the agreed results on time with great quality.
- The Scrum Master is not responsible for keeping his team organized that is they it is the duty of the Scrum Team to get self-organized.
- They have to be agile in the office and have to attend every stand-up and other ceremonies.
- They have to participate in all the meetings despite their nature and have to ensure that all the findings of the meetings are getting practically addressed in the project.

Stakeholders:

- The Stakeholder has to keep a healthy relationship with the Product Owner in order to share every detail regarding his project.
- The Stakeholder is responsible for conveying his wishes and concerns to the product owner or else the product owner would not be responsible for his project quality and time duration.
- The Stakeholder has to provide regular input to queries from the Product Owner.
- Prioritizing the work effectively with the Product Owner is another job that the Stakeholder has to do to ensure his project development.
- Keep taking updates or keep giving updates regarding any change in the plans.

What are Agile Ceremonies?

Sprint Planning

Sprint Planning is used to determine what the team will accomplish in the upcoming Sprint. The event itself has two parts. The first half of Sprint Planning is used to determine 'What' the team will be working on, by pulling items from their Backlog into their Sprint Backlog. The second half of Sprint Planning is when the Development Team determines 'How' they will accomplish the work that's been pulled into the Sprint Backlog.

Sprint Review

The Sprint Review is when the team presents their work from the Sprint to the project's stakeholders. It should cover not only the work they accomplished, but also open discussions around the work they were not able to complete. The attendees of this event should include anyone with a vested interest in the project. Particularly stakeholders, clients, and end-users.

Sprint Retrospective

The Sprint Retrospective is the primary event in which the Scrum Team can inspect and adapt their approaches based on their experiences from the previous sprints. Retrospectives can be held using a large variety of games, questions, and exercises; but at it's core, the Sprint Retrospective helps the team to determine: What worked well in the last sprint? What did not work well? And what can be implemented into the next Sprint to improve how the Scrum Team does it's work? Retrospectives allow the team to consistently improve from one Sprint to the next.

Daily Scrum

The Daily Scrum, sometimes referred to as the Daily Stand-up, has a time-box for 15 minutes or less, and is specifically for the benefit of the development team. The goal of this event is for the team to get in

sync on a daily basis, allowing for better collaboration and transparency. The Daily Scrum should be held at the same time each day and should not include anyone outside of the Scrum Team. Traditionally, the Daily Scrum involves each team member answering three questions:

- What did I achieve yesterday to help us meet our Sprint Goal?
- What do I hope to achieve today to help us meet our Sprint Goal?
- Do I see any impediments that prevent me or my team from achieving our Sprint Goal?

What are Agile Artifacts?

Illustration 23: Agile Artifacts

Increment

The Increment is the sum of all the Product Backlog items completed during a Sprint and all previous Sprints.

At the end of a Sprint, the new Increment must be "Done," which means: It must meet the Scrum Team's Definition of "Done."

Product Backlog

A product backlog is a list of all the things that are required in the product and it is a dynamic and best understood requirements for any changes to be made to the product. Product backlog owned by the Product Owner (PO) which consists of a lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in the future releases.

Sprint Backlog

The Sprint Backlog is the set of Product Backlog items selected for the Sprint plus a plan for delivering the product Increment and realizing the Sprint Goal. The Sprint Backlog is a forecast by the Development Team about what functionality will be in the next Increment and the work needed to deliver that functionality. The Sprint Backlog defines the work the Development Team will perform to turn Product Backlog items into a "Done" Increment. The Sprint Backlog makes visible all of the work that the Development Team identifies as to meet the Sprint Goal.

What is Agile Tools?

The list below shows some of the best tools on offer. For a complete list, see this post.

- Active Col lab

An affordable tool for small businesses, Active Col lab is easy to use. This software development aid requires little training and provides excellent support.

- Agilo for Scrum

Stakeholders get updated automatically on the project's progress with Agile for Scrum. Features sprint reports and burn down charts for better data mining.

- Atlassian Jira + Agile

This powerful project management tool facilitates development by incorporating Scrum, Kanban, and customization workflows.

- Pivotal Tracker

This methodology tool is geared specifically for mobile projects. A little jargon-heavy, it's user-friendly after a brief orientation period.

- Prefix

This free tool from Stackify provides an instant feedback loop to catch and fix bugs before they can deploy.

- Retrace

For a more robust solution complete with monitoring, errors, logs, and more, Stackify's Retrace provides app performance insights from integration to QA to production, at the code level.

REFERENCES

[1]. The below link is used for drawing the UML diagrams

https://app.diagrams.net

[2] https://lucid.app.

[3] https://www.atlassian.com/agile

[4] https://www.youtube.com/watch?v=WjwEh15M5Rw

[5] https://www.youtube.com/watch?v=oTZd2vo3FQU&t=337s