./

Report – Change Detection in Image Sequence

Course Code: <CODE>



Version Number:

Team Members :

Team No:

Module: Model Based System Engineering

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **Approved By** | **Remarks/Revision Details** |
| 1.0 | 03-12-2020 | Prathviraj V | 99003124  99003125 | Srinivas K |  |
| 1.1 | 05-12-2020 | Prathviraj V | 99003122  99003121 | Srinivas K |  |
| 1.2 | 07-12-2020 | Prathviraj V | 99003122  99003124 | Srinivas K |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

# 

Contents

[ACTIVITY 1 5](#_Toc58332813)

[PROBLEM STATEMENT: 5](#_Toc58332814)

[INTRODUCTION: 5](#_Toc58332815)

[REQUIREMENTS 6](#_Toc58332816)

[HIGH LEVEL REQUIREMENTS 6](#_Toc58332817)

[LOW LEVEL REQUIREMENTS 6](#_Toc58332818)

[SWOT 6](#_Toc58332819)

[SURVEY 7](#_Toc58332820)

[TEST PLAN 8](#_Toc58332821)

[DIAGRAMS 10](#_Toc58332822)

[BEHAVIORAL DIGAGRAM 10](#_Toc58332823)

[STRUCTURAL DIAGRAM 12](#_Toc58332824)

[ACTIVITY 2 13](#_Toc58332825)

[AGILE ASPECTS 13](#_Toc58332826)

[EPIC 13](file:///C:\Users\test.DESKTOP-91D4PP6\Downloads\Individual_project_report1.2.docx#_Toc58332827)

[THEME 13](file:///C:\Users\test.DESKTOP-91D4PP6\Downloads\Individual_project_report1.2.docx#_Toc58332828)

[USER STORY 13](file:///C:\Users\test.DESKTOP-91D4PP6\Downloads\Individual_project_report1.2.docx#_Toc58332829)

[BACKLOGS: 14](#_Toc58332830)

[RTM (Requirement Traceability Matrix) 14](#_Toc58332831)

[ACTIVITY 3 15](#_Toc58332832)

[CI workflows for Genesis 15](#_Toc58332833)

[GitHub Badges 15](#_Toc58332834)

[GitHub Repository 16](#_Toc58332835)

[COMMITS 17](#_Toc58332836)

[ACTIONS 18](#_Toc58332837)

[BUILDS 18](#_Toc58332838)

[ISSUES 19](#_Toc58332839)

[ACTIVITY 4 20](#_Toc58332840)

[GANTT CHART 20](#_Toc58332841)

[REFERENCES 21](#_Toc58332842)

**Table of Tables**

[Table 1:High Level Requirements 6](#_Toc58329541)

[Table 2:Low Level Requirements 6](#_Toc58329542)

[Table 3: SWOT Analysis 6](#_Toc58329543)

[Table 4: Test Plan 8](#_Toc58329544)

[Table 5: RTM 14](#_Toc58329545)

**Table of Figures**

[Figure 1:Activity Diagram 10](#_Toc58242012)

[Figure 2: Use Case Diagram 11](#_Toc58242013)

[Figure 3:Component Diagram 12](#_Toc58242014)

[Figure 4:Class Diagram 12](#_Toc58242015)

[Figure 5: Agile Aspects 14](#_Toc58242016)

[Figure 6: GitHub Badges 15](#_Toc58242017)

[Figure 7: Repository Structure 1 16](#_Toc58242018)

[Figure 8: Repository Structure 2 16](#_Toc58242019)

[Figure 9: Commit 2 17](#_Toc58242020)

[Figure 10: Commit 1 17](#_Toc58242021)

[Figure 11: GitHub Actions 18](#_Toc58242022)

[Figure 12: GitHub Build 18](#_Toc58242023)

[Figure 13: Code Quality 19](#_Toc58242024)

[Figure 14: Issues 19](#_Toc58242025)

[Figure 15: Gantt Chart 20](#_Toc58242026)

# ACTIVITY 1

## PROBLEM STATEMENT:

To develop a system that eﬃciently detects the changes in a video sequence in which the camera moves in a deﬁned path that is given as input and compare the objects in the second video in which the camera moves in the same path, to detect the new and missing objects showing bounding boxes around it and can be deployed in a parking lot**.**

## INTRODUCTION:

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Well-researched domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance. It is widely used in computer vision tasks such as image annotation, activity recognition, face detection, face recognition, video object co-segmentation. It is also used in tracking objects, for example tracking ball during a football match, tracking movement of a cricket bat, or tracking a person in a video.

Methods for object detection commonly classiﬁed into two approaches either machine learning-based approaches or deep learning-based approaches. In the Machine Learning approaches, it is very necessary to ﬁrst specify the all features using one of the methods, then using a technique such as support vector machine (SVM) to do the classiﬁcation. The one of the technique deep learning techniques are able to do end-to-end object detection without speciﬁcally deﬁning features, and they are typically based on convolution neural networks (CNN).

Object tracking is the one of the process in the tracking system of locating a moving object (or multiple objects) over the time using a speciﬁc camera. It has a several of uses, some of the main uses are: human-computer interaction, security and surveillance, video communication and compression, augmented reality, traﬃc control, medical imaging and video editing.

Machine learning approaches:

* Viola–Jones object detection framework based on Haar features.
* Scale-invariant feature transform (SIFT).
* Histogram of oriented gradients (HOG) features.

Deep learning approaches:

* Region Proposals (R-CNN, Fast R-CNN, Faster R-CNN.
* Single Shot MultiBox Detector (SSD).
* You Only Look Once (YOLO).
* Single-Shot Reﬁnement Neural Network for Object Detection (ReﬁneDet).
* Retina-Net.
* Deformable convolution networks.

## REQUIREMENTS

### HIGH LEVEL REQUIREMENTS

|  |  |
| --- | --- |
| **ID** | **Description** |
| HL\_01 | Must upload the reference video |
| HL\_02 | Must upload the new video |
| HL\_03 | Operating System: Windows 10 (64bit) |
| HL\_04 | Image Processing Frameworks: OpenCV, skimag |
| HL\_05 | Other packages: numpy, scipy, tkinter, PIL, multiprocessing |
| HL\_06 | System with sufficient RAM to run the project |

Table : High Level Requirements

### LOW LEVEL REQUIREMENTS

|  |  |
| --- | --- |
| **ID** | **Description** |
| LL\_01 | Must store the record in the output file |
| LL\_02 | Must view the entire output file |
| LL\_03 | Must match the correct objects in both videos |
| LL\_04 | Must have a friendly GUI |
| LL\_05 | Must categorize all the detected objects |
| LL\_06 | Must run for videos in slight different path |
| LL\_07 | Must update regularly |

Table : Low Level Requirements

## SWOT

|  |  |  |  |
| --- | --- | --- | --- |
| **Strength** | **Weakness** | **Opportunity** | **Threat** |
| Useful for detecting missing and new objects | The detection is time consuming | Can be used in many scenarios wherever required |  |
| We can reduce thefts | We won’t be able to detect thief if the video is not recorded at that time | We will be able to catch the thieves easily with the face detection techniques | It is very sensitive in case of non-clear videos |
| We can implement in defense | Maintenance cost will be high |  | The accuracy will reduce when the video path changes |

Table : SWOT Analysis

## SURVEY

**6.1** **MULTIPLE REAL-TIME OBJECT IDENTIFICATION USING SINGLE SHOT MULTI-BOX DETECTION**

**Authors: Kanimozhi S, Gayathri G, Mala T.**

Continuous item location is one of the difficult task as it need quicker calculation power in distinguishing the item around then. Anyway, the information produced by any ongoing framework are unlabeled information which regularly need enormous arrangement of named information for powerful preparing reason. This paper proposed a quicker location technique for constant item discovery dependent on convolution neural system model called as Single Shot Multi-Box Detection(SSD). This work takes out the component resampling arrange and joined every determined outcome as a solitary segment. Still there is a need to find a light weight arrange model for the spots which needs computational force like cell phones (e.g.: PC, cell phones, and so on). Along these lines a light weight organize model which use profundity insightful distinguishable convolution called Mobile Net is utilized right now. Trial result uncover that utilization of Mobile Net alongside SSD model increment the precision level in distinguishing the continuous family questions.

**6.2 YOLOV3: AN INCREMENTAL IMPROVEMENT**

**Authors: Joseph Redmon, Ali Farhadi University of Washington.**

We present a few updates to jYOLO! We made a pack of little structure changes to improve it. We additionally prepared this new system that is entirely swell. It's somewhat greater than last time however increasingly exact. It's despite everything quick however, don't stress. At 320 × 320jYOLOv3 runs in 22ms at 28.2mAP, as precise asjSSD however multiple times quicker. At the point when we look at the old .5IOUmAP recognition metric YOLOv3 is very great. It accomplishes 57.9AP50 in 51ms on a Titan X, compared to 57.5 AP50 in 198ms by Retina Net, comparative execution yet 3.8X quicker.

**6.3 MOVING OBJECT DETECTION AND TRACKING USING CONVOLUTIONAL NEURAL NETWORKS**

**Authors: ShraddhajMane and Prof. SupriyajMangale.**

The item recognition and following is the significant steps of PC vision calculation. The hearty item location is the test because of varieties in the scenes. Another greatest challenge is to follow the article in the impediment conditions Tensor Flow object identification API. An epic CNN based object following calculation is utilized for hearty item discovery. The proposed approach can distinguish the article in various light and impediment. The proposed approach accomplished the exactness of 90.88% on self-produced picture successions.

**6.4 IMAGE CHANGE DETECTION ALGORITHMS: A SYSTEMATIC SURVEY**

**Authors: Richard J. Radke, Srinivas Andra, Omar Al-Kofahi, and Badrinath Roysam**

Identifying districts of progress in numerous pictures of the same scenerio taken at various occasions is of far reaching enthusiasm due to an enormous number of utilizations in diﬀerent controls, including remote detecting, reconnaissance, clinical analysis and treatment, common foundation, and submerged detecting. This paper speciﬁes a precise review of the normal handling steps and center choice principles in present day change recognition calculations, including centrality and speculation testing, prescient models, the concealing model, and foundation displaying. We additionally talk about signiﬁcant pre preparing strategies, ways to deal with authorizing the regularity of the change veil, and standards for assessing and looking at the exhibition of progress recognition calculations. It is trusted that our order of calculations into a generally modest number of classes will give helpful direction to the calculation architect.

## TEST PLAN

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Precondition** | **Expected I/P** | **Expected O/P** | **Actual O/P** |
| HL\_01\_T\_01 | Upload the reference video in given format | System is running | Reference video | Reference video is uploaded |  |
| HL\_02\_T\_02 | Reference video must be formatted properly | System is running | Reference video in specified format | Reference video is formatted properly |  |
| HL\_03\_T\_03 | Reference video must be converted into image frames | Reference video is formatted properly | Reference video | Video is converted to image frames |  |
| HL\_03\_T\_04 | Upload the new video in given format | System is running | New video | New video is uploaded |  |
| LL\_01\_T\_05 | New video must be converted into image frames | System is running | New video | New Video is converted to image frames |  |
| LL\_02\_T\_06 | Must detect the objects in the reference video | Objects are not detected | Image frames of reference video | All objects in video are detected and given unique ID |  |
| LL\_03\_T\_07 | Must detect the objects in the new video | Objects are not detected | Image frames of new video | All objects in video are detected and given unique ID |  |
| LL\_03\_T\_08 | Must match the objects in the reference video with the objects in the new video | Objects in both the videos are detected | Detected objects with unique Id | Common objects in reference video and new videos must be matched with each other |  |
| LL\_04\_T\_09 | Has to track the missing and the new objects | Common objects in both the videos are matched | Objects that are matched | Find the new and missing objects in the video |  |
| LL\_05\_T\_10 | Find the approximate location of the new and missing objects in the new video | New and missing objects are detected | Object ID of the new and missing objects | Appropriate location of the new and missing objects |  |
| LL\_06\_T\_11 | Add bounding boxes around the new and missing objects | Location of the new and missing objects is known | Timeframe of new and missing objects | Bounding boxes around new and missing objects |  |

Table : Test Plan

## DIAGRAMS

### BEHAVIORAL DIGAGRAM

* **Activity Diagram**

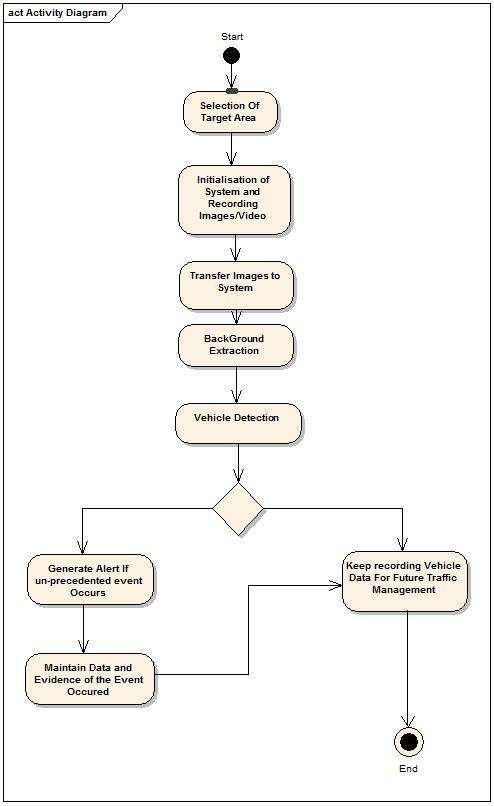


Figure 1:Activity Diagram

* **Use case Diagram**

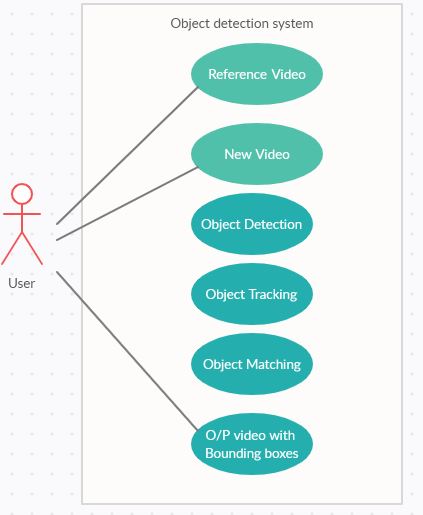


Figure 2: Use Case Diagram

### STRUCTURAL DIAGRAM

* **Component Diagram**

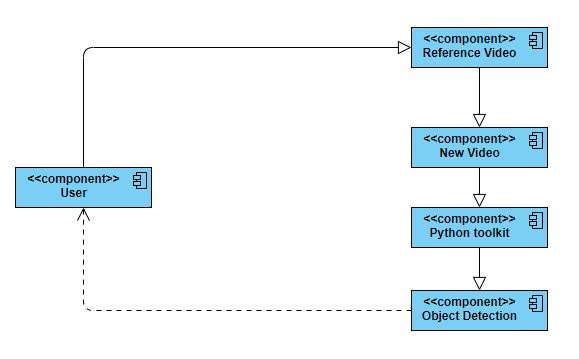
****

Figure 3:Component Diagram

* **Class Diagram**

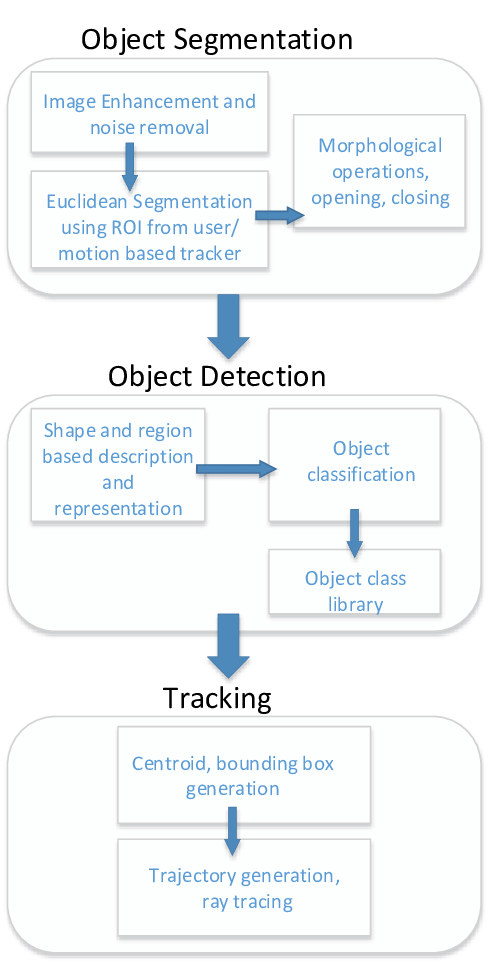


Figure 4:Class Diagram

# ACTIVITY 2

## AGILE ASPECTS

**USER STORY**

The system is unable to track the exact location of the new or the missing objects

**USER STORY**

The objects in reference video or the new video might not have been detected properly

**USER STORY**

The objects in the new videos might have been matched with different objects in the reference video

### EPIC

The system should detect new and missing objects

THEME  
“To detect the new and missing objects in the given video sequence

**EPIC**

The system is unable to detect the new or missing objects

### USER STORY

The system should add bounding boxes around the new and missing objects

### BACKLOGS:

### PRODUCT BACKLOGS:

1. When the videos are being of moving objects then the detection and tracking of objects will not be accurate if the video is recorded faster
2. When the video is not recorded in full clarity and there are lot of noise in that than the detection will be difficult
3. When there are many similar objects then it is difficult match the similar objects
4. When the path of the video changes then the object tracking will be difficult

### SPRINT BACKLOG:

1. The path of the vehicles should be detected when we give the video of the road is given in case of DRDO – Week 1

### RTM (Requirement Traceability Matrix)

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement No** | **Requirement Description** | **Testcase ID** | **Status** |
| RN\_01 | Input the reference and new video | TC\_01  TC\_02  TC\_03 | PASS  PASS  FAIL |
| RN\_02 | Detect objects in both the videos | TC\_04  TC\_05  TC\_06 | PASS  PASS  FAIL |
| RN\_03 | Object tracking | TC\_07  TC\_08 | PASS  NO RUN |
| RN\_04 | Find the new and missing objects | TC\_09  TC\_10  TC\_11 | PASS  FAIL  NO RUN |

Table : RTM

### DESIGN

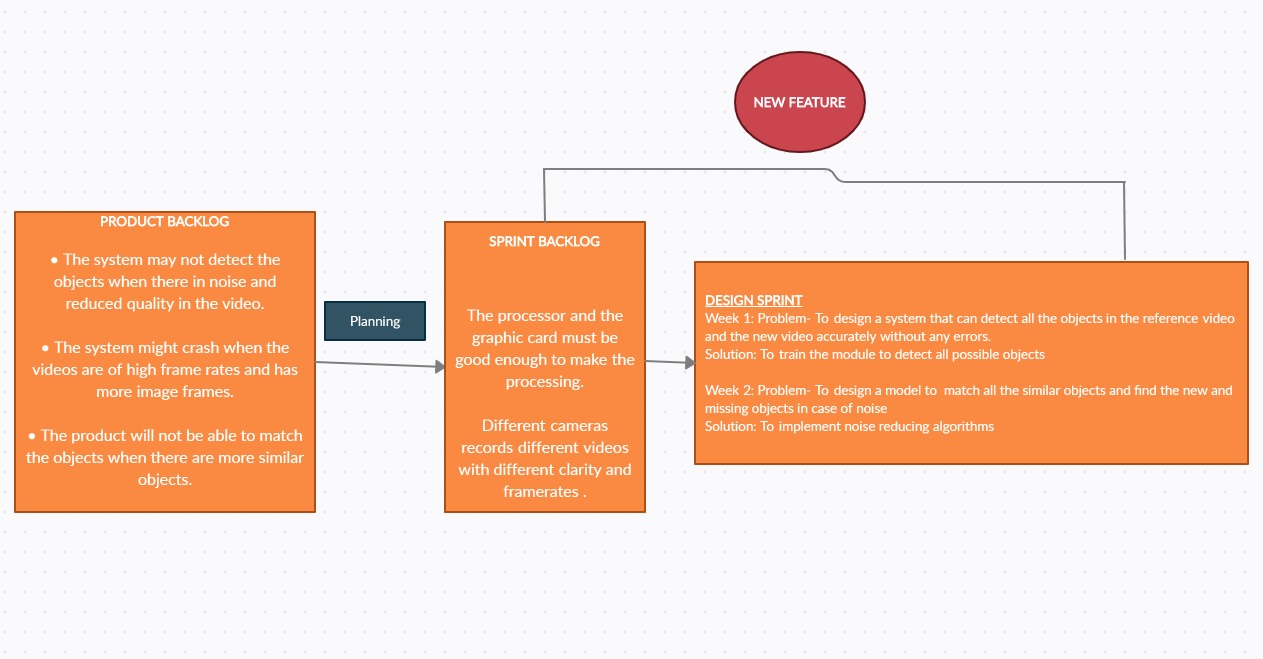


Figure 5: Agile Aspects

# ACTIVITY 3

## CI workflows for Genesis

GITHUB REPOSITORY LINK

### GitHub Badges

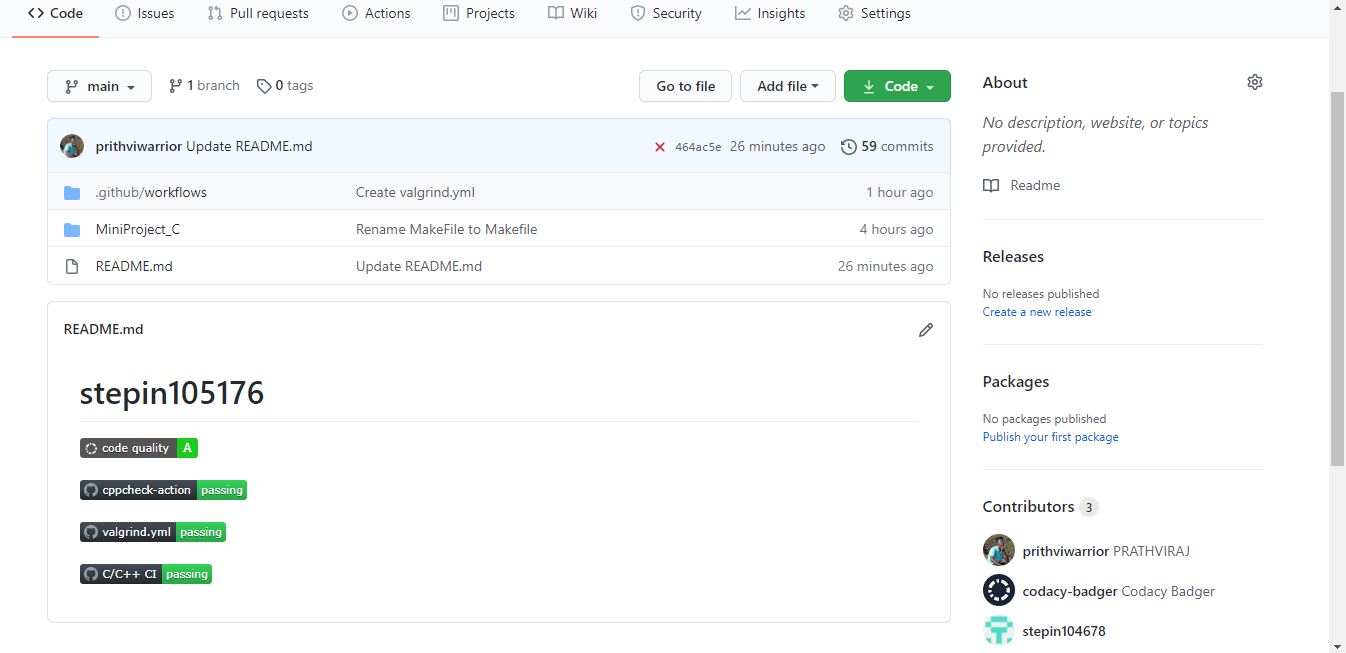


Figure 6: GitHub Badges

### GitHub Repository

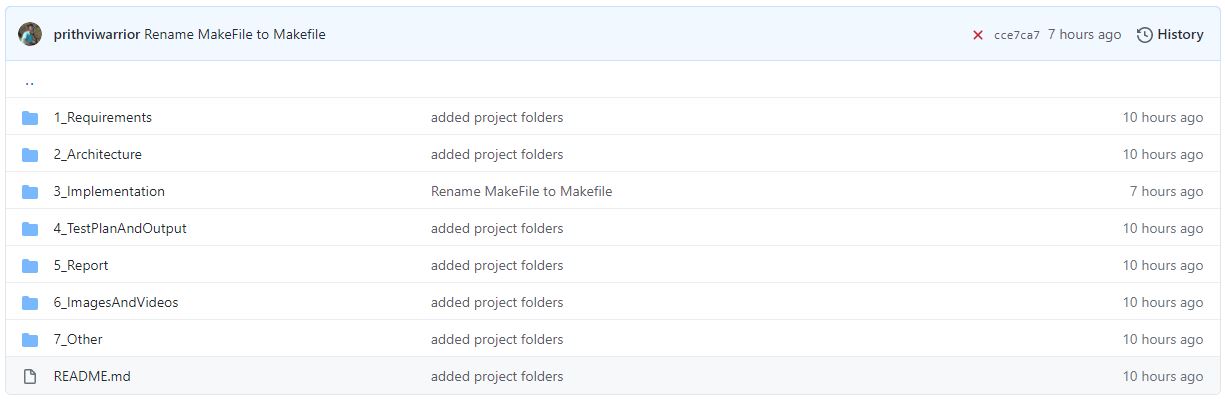


Figure 7: Repository Structure 1

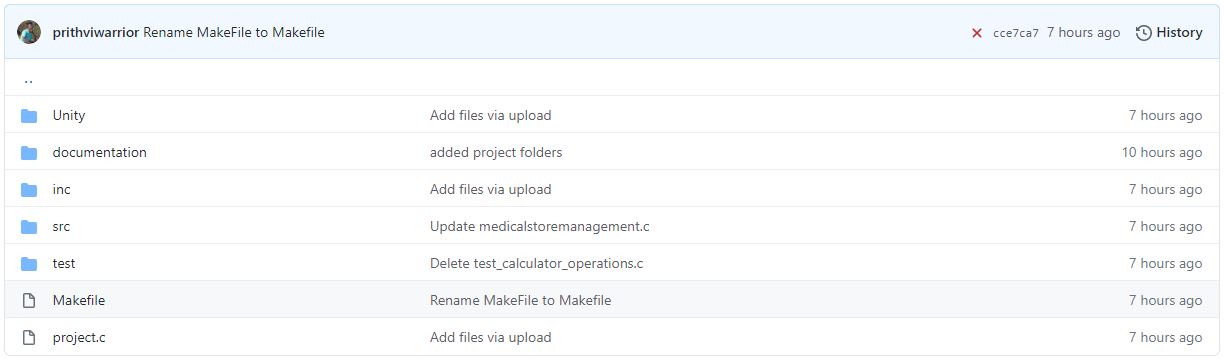


Figure 8: Repository Structure 2

### COMMITS

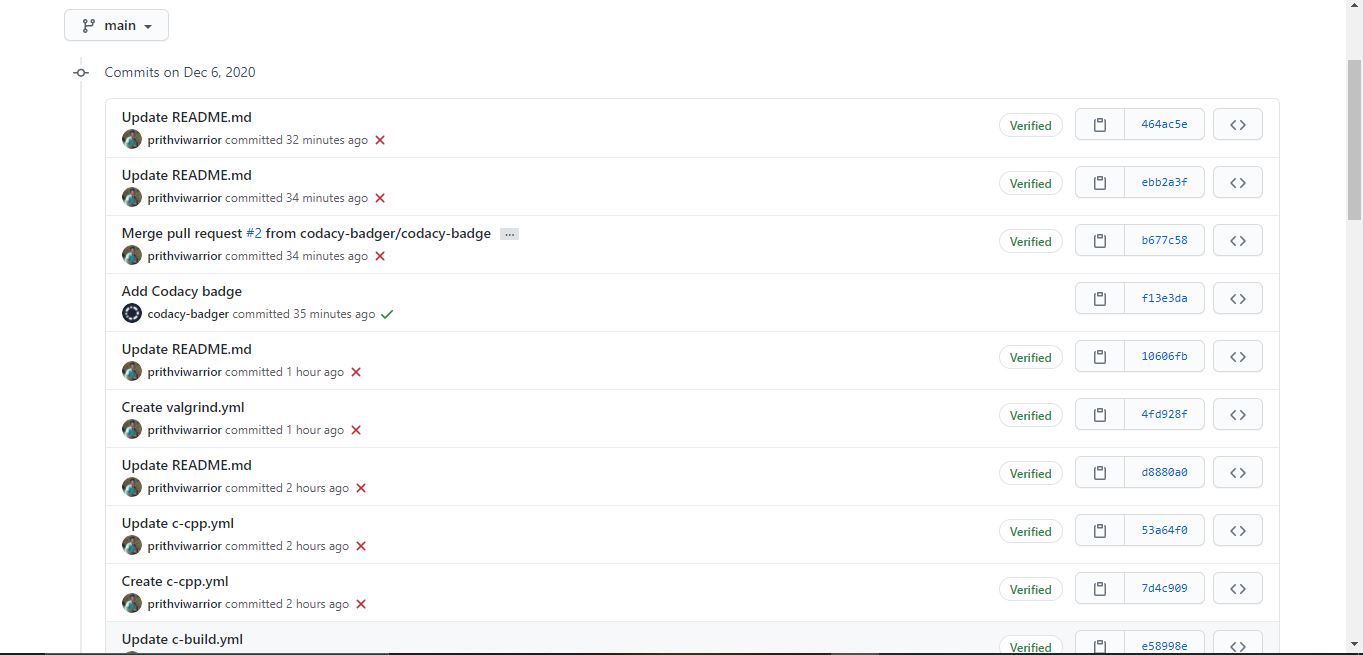


Figure 9: Commit 2

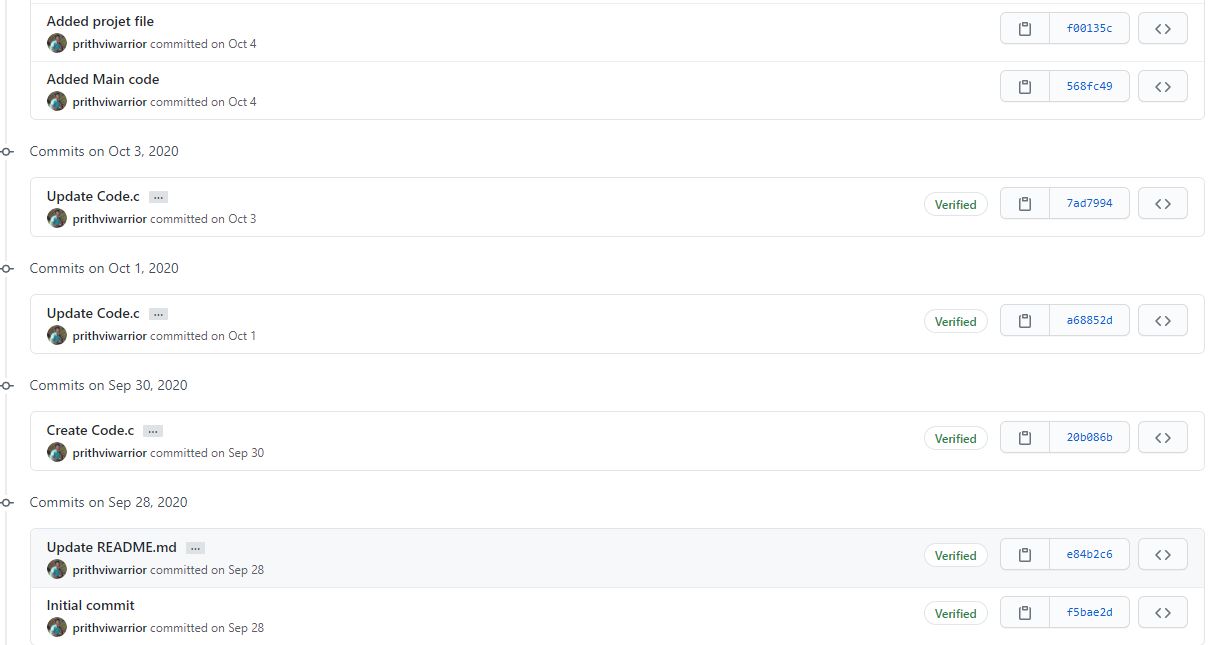


Figure 10: Commit 1

### ACTIONS

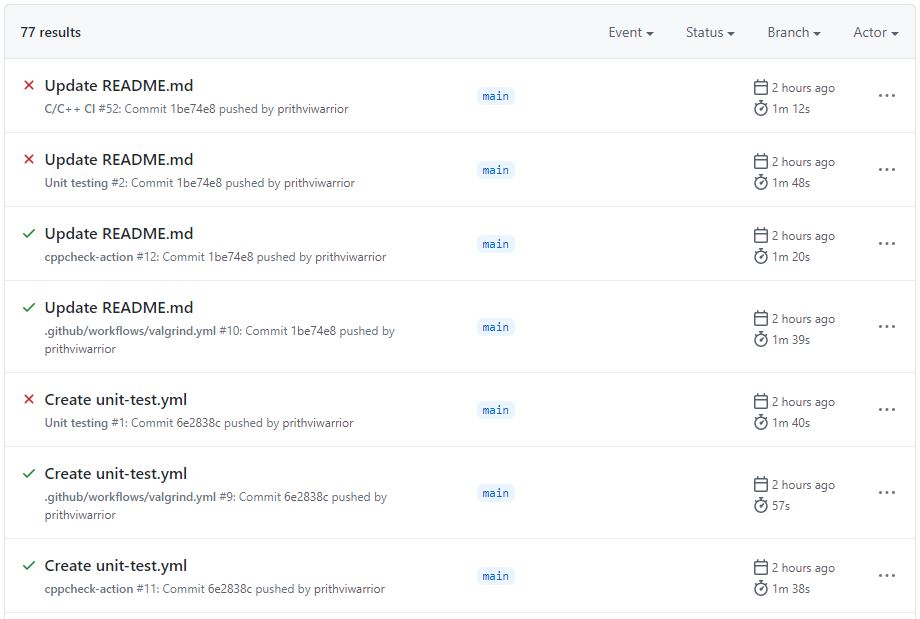


Figure 11: GitHub Actions

### BUILDS

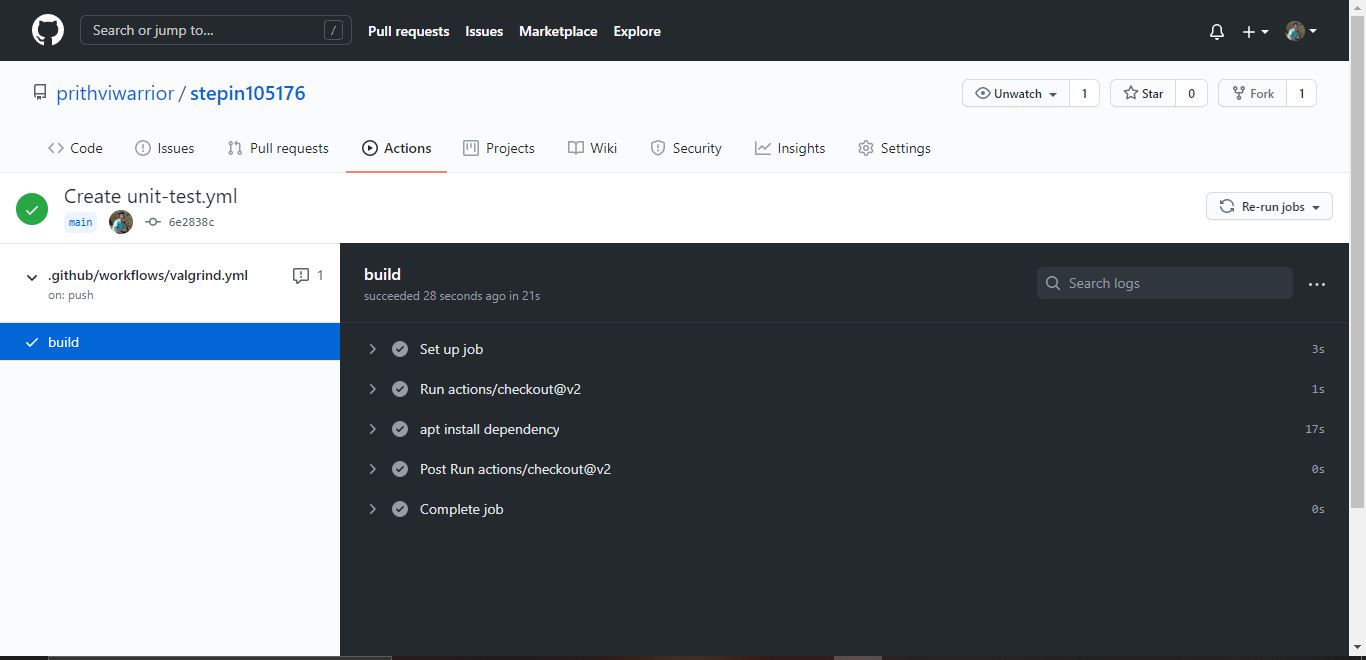


Figure 12: GitHub Build

### ISSUES

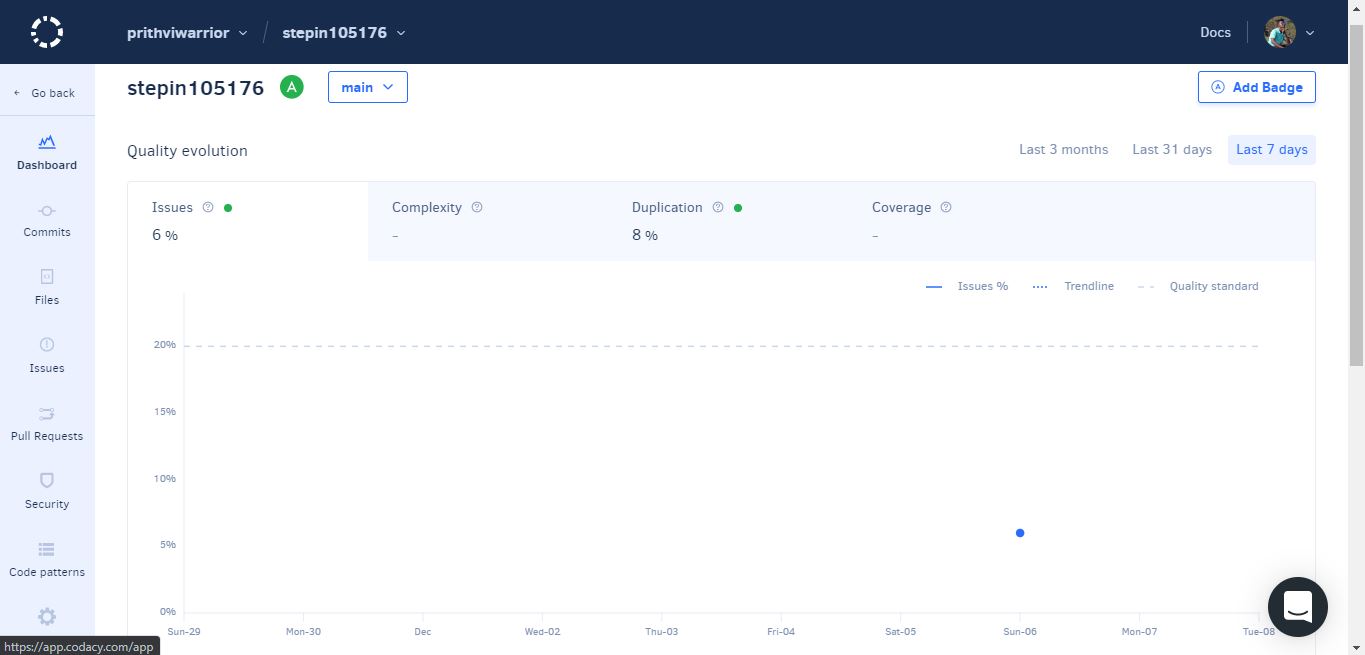


Figure 13: Code Quality

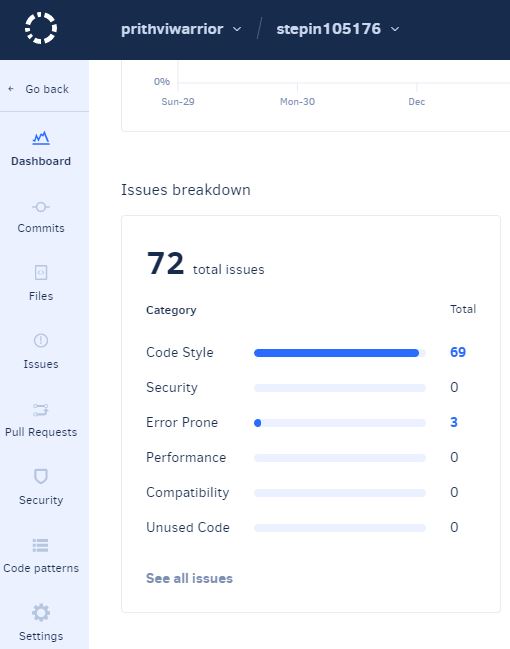
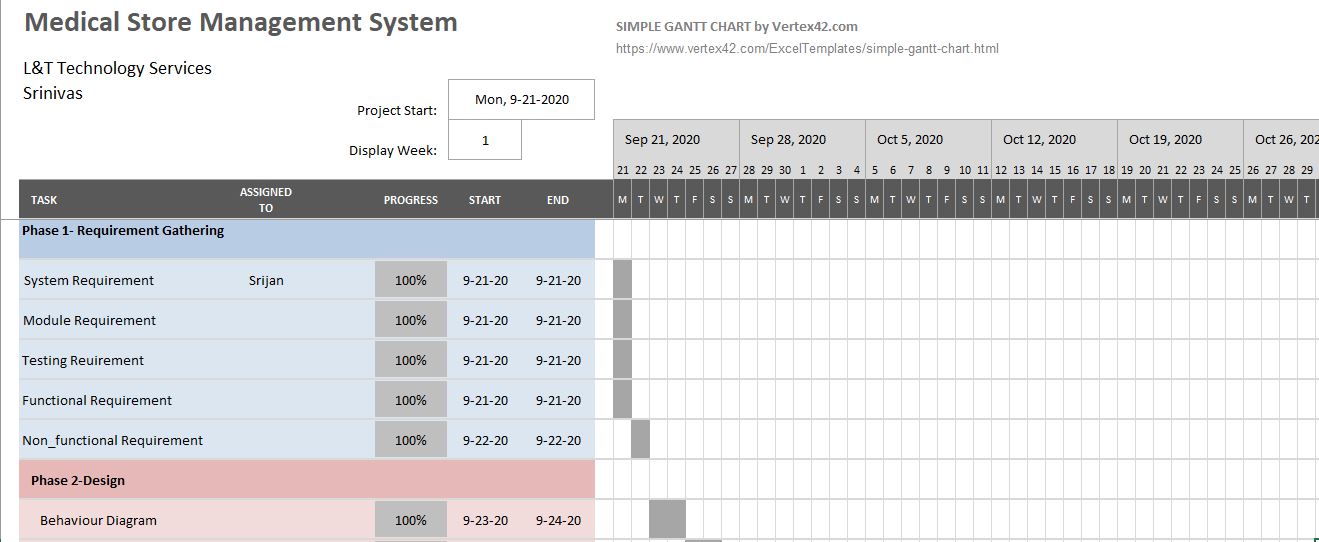
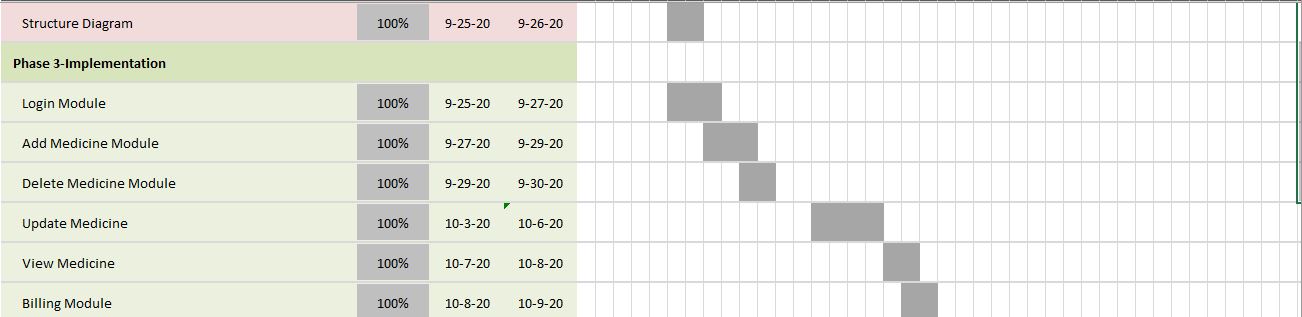


Figure 14: Issues

# ACTIVITY 4

## GANTT CHART





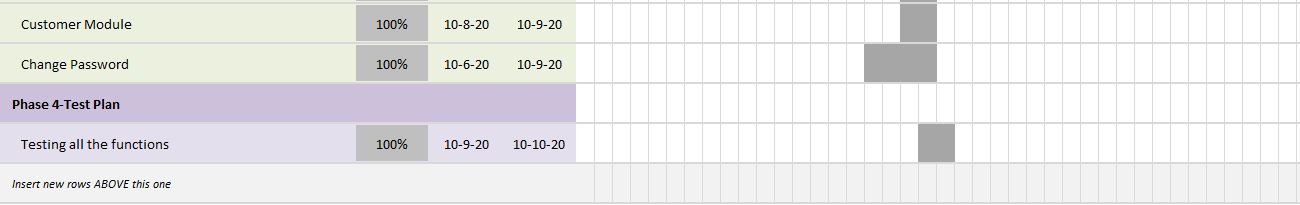


Figure 15: Gantt Chart

## REFERENCES

1. MULTIPLEjREAL-TIME OBJECTjIDENTIFICATION USING SINGLE SHOT MULTI-BOXjDETECTION

Authors: Kanimozhi S, Gayathri G, Mala T

1. YOLOV3: AN INCREMENTAL IMPROVEMENT

Authors: JosephjRedmon, Ali FarhadijUniversity of Washington.

1. MOVING OBJECTjDETECTION AND TRACKING USING CONVOLUTIONAL NEURALjNETWORKS

Authors: ShraddhajMane and Prof. SupriyajMangale.

1. IMAGE CHANGE DETECTION ALGORITHMS: A SYSTEMATIC SURVEY

Authors: Richard J. Radke, Srinivas Andra, Omar Al-Kofahi, and Badrinath Roysam

1. [Object Tracking using OpenCV (C++/Python) | Learn OpenCV](https://www.learnopencv.com/object-tracking-using-opencv-cpp-python/#:~:text=%20What%20is%20Object%20Tracking%3F%20%201%20Dense,function.%20They%20are%20also%20used%20for...%20More%20)
2. [What Is Object Detection? - MATLAB & Simulink (mathworks.com)](https://in.mathworks.com/discovery/object-detection.html)
3. https://app.creately.com/diagram/ih4Eva9d4kk/edit