

# GOOD MORNING!

早上好!

안녕하세요!

---

DAY 4

# DAY I (DONE)

---

- Welcome
- Project Introduction
- Introduction to Project Development Process
- Business Requirement Development
- System Requirement Development
- System(High Level) Design
- Time Management

# DAY 2 (DONE?)

---

- YOLOv8 기반 데이터 수집/학습/deploy (Detection Alert)
  - 감시용 데이터 수집(bus, truck, tank 등)
  - 감시용 데이터 라벨링
  - YOLOv8 기반 학습
  - YOLOv8 Object Detection
- Porting to ROS
  - Create Detection Alert Node
  - Generate Topics to send image and Obj. Det. results
  - Create Subscriber node and display image and print data from the Topic

# DAY 3 (DONE?)

---

- AMR (Autonomous Mobile Robot) 기반 카메라 인식 autonomous driving 시스템 with obstacle avoidance 구축 (AMR Controller)
  - Digital Mapping of environment
  - Goal Setting and Obstacle Avoidance using Navigation
  - Object Tracking w/ AMR camera
  - Control logic between navigation/obj. tracking/ obj. following (teleop)
- Porting to ROS
  - Create AMR Controller Node
  - Create and send Obj.Tracking Image and data to Sysmon
- And finally, Integration and Test of Detection Alert & AMR Controller

# DAY 4

---

- Flask 를 이용한 웹 서버 구축 (System Monitor)
  - Flask/HTML Intro
  - Deploy YOLOv8 Obj. Det results to web
  - Log in 기능 구현
  - Sysmon 웹기능 구현
  - 알람 기능 구현
- SQLite3를 이용한 데이터베이스 구축 및 연동 (System Monitor)
  - SQLite3 기본 기능 구현
  - DB 기능 구축
  - 알람이 울리는 경우 DB에 저장하는 기능 구현
  - 저장된 내용 검색하는 기능 구현



# DAY 4

---

- Porting to ROS
  - Update Sysmon Node code
  - Update the database with received Obj. Det. Data from Detection Alert Node
  - Display the content of DB on System Monitor web page
- And finally, Integration and Test of Detection Alert, AMR Controller & System Monitor

# DAY 5

---

- 감시시스템 통합 구현
  - - 전체 시스템 통합 운용
- Team Demo & Presentation
- 평가 시간

프로젝트 RULE NUMBER ONE!!!

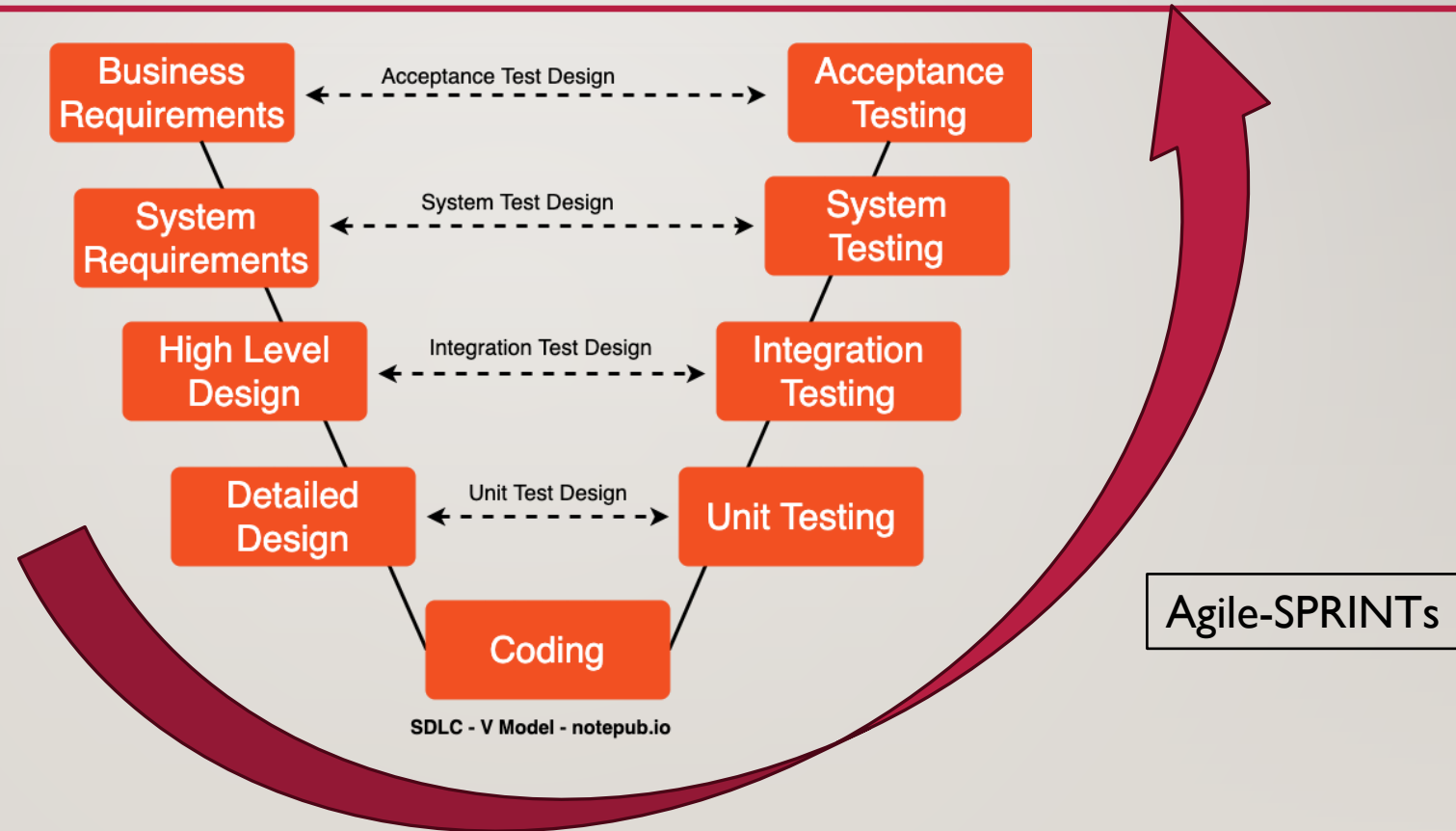
---

Have Fun Fun Fun!

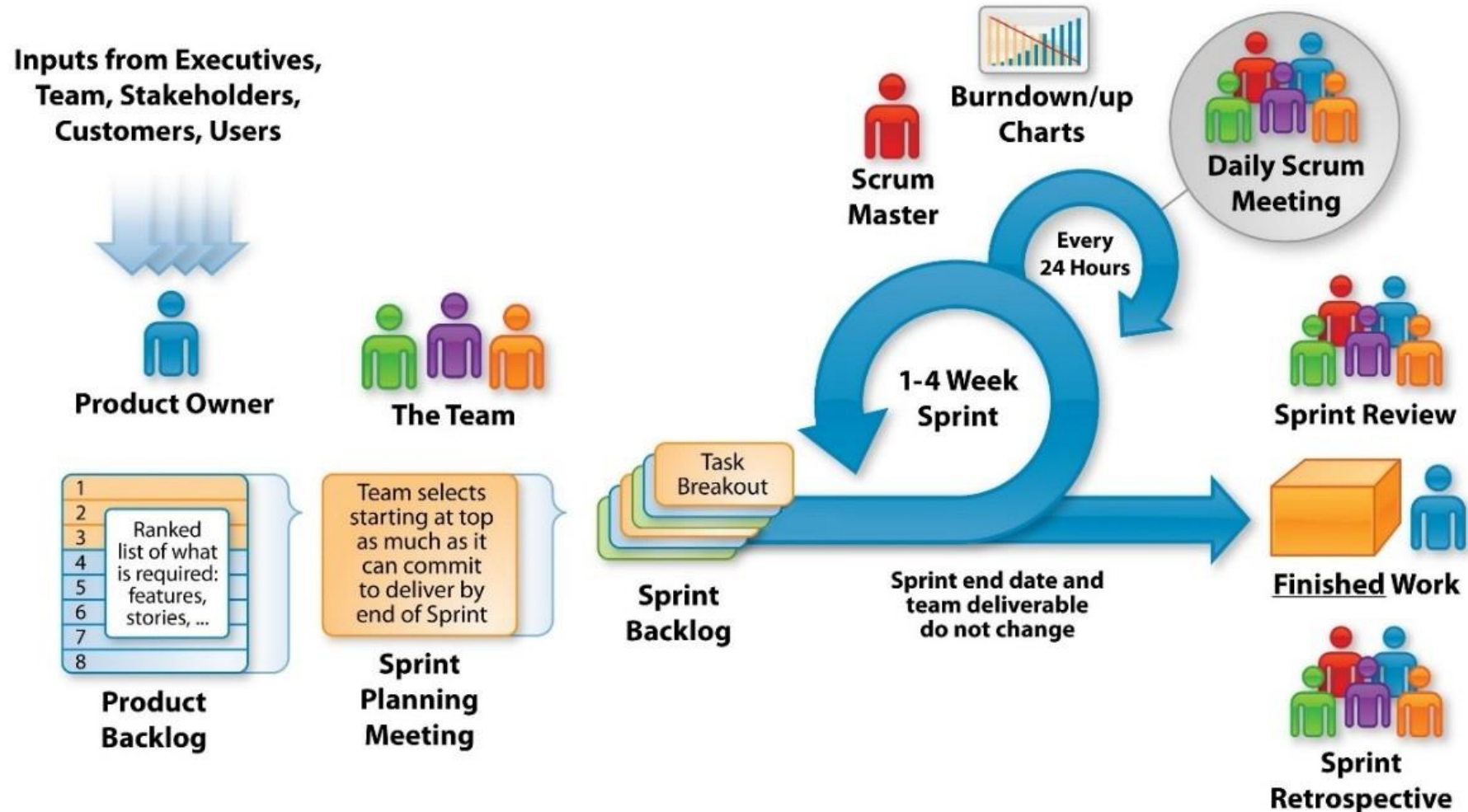




# SW DEVELOPMENT PROCESS



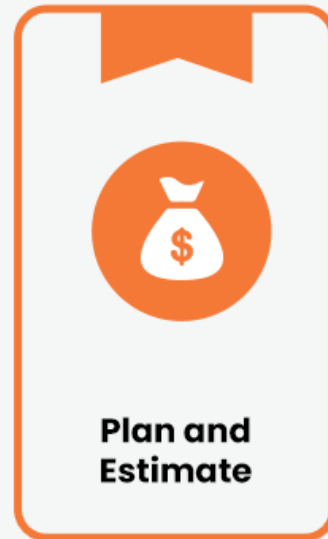
# The Agile - Scrum Framework



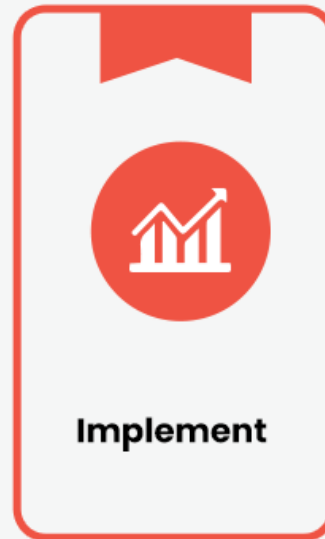
# 5 Stages of Scrum Sprint



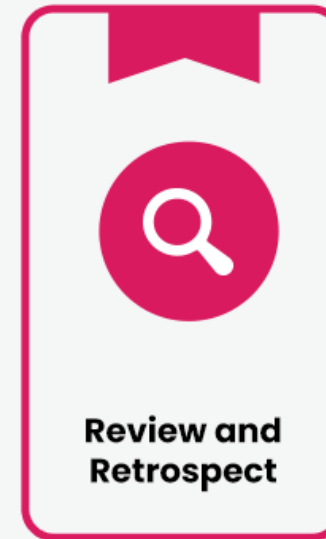
This phase includes the processes related to the commencement of a project, such as a scope and objectives, creating and distributing its charter, and taking other steps to guarantee success.



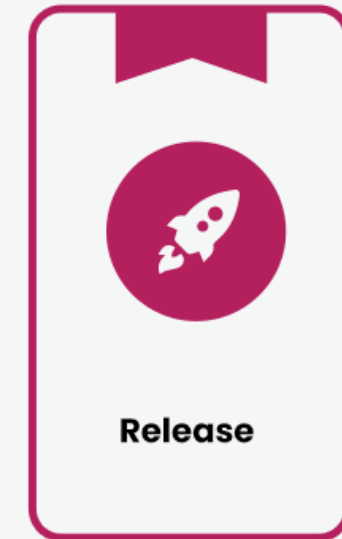
This phase involves planning and estimating processes, including creating user stories, approving, assessing, committing user stories, creating tasks, evaluating tasks, and creating a Sprint backlog.



This phase is about executing the tasks and activities to create a product. These activities include building the various outputs, conducting daily standup meetings, and grooming the product backlog.

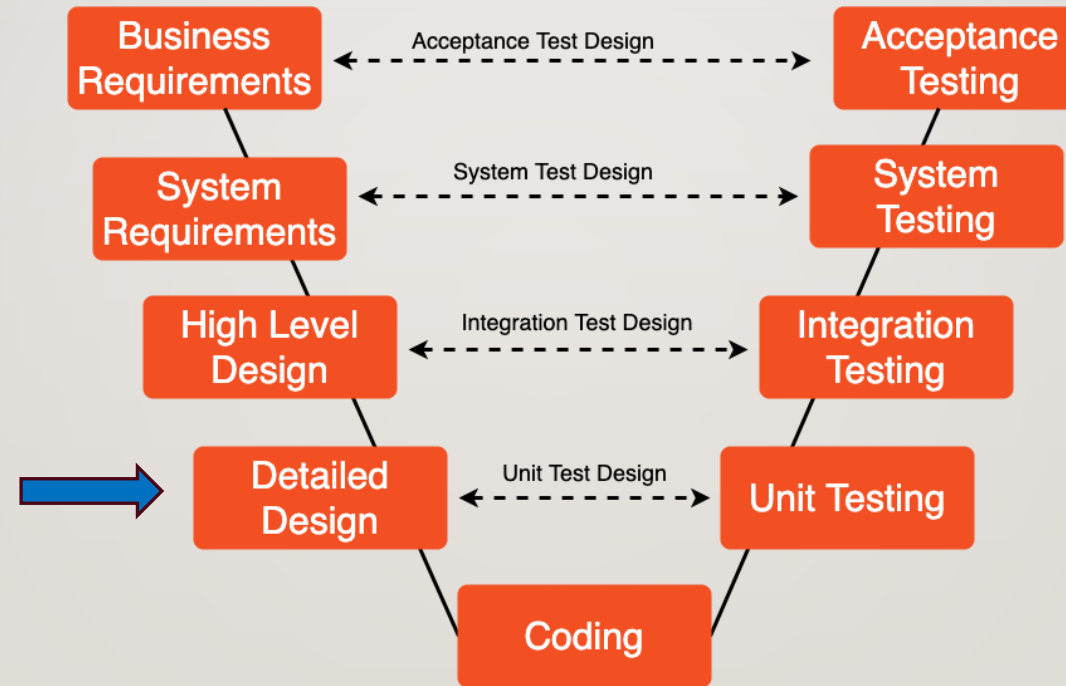


This stage of the project lifecycle is concerned with evaluating what has been accomplished so far, whether the team has worked to plan, and how it can do things better in the future.



This stage highlights delivering the accepted deliverables to the customer and determining, documenting, and absorbing the lessons learned during the project.

# SPRINT 2 – AMR CONTROLLER



SDLC - V Model - notepub.io



# TEAM EXERCISE 6

---

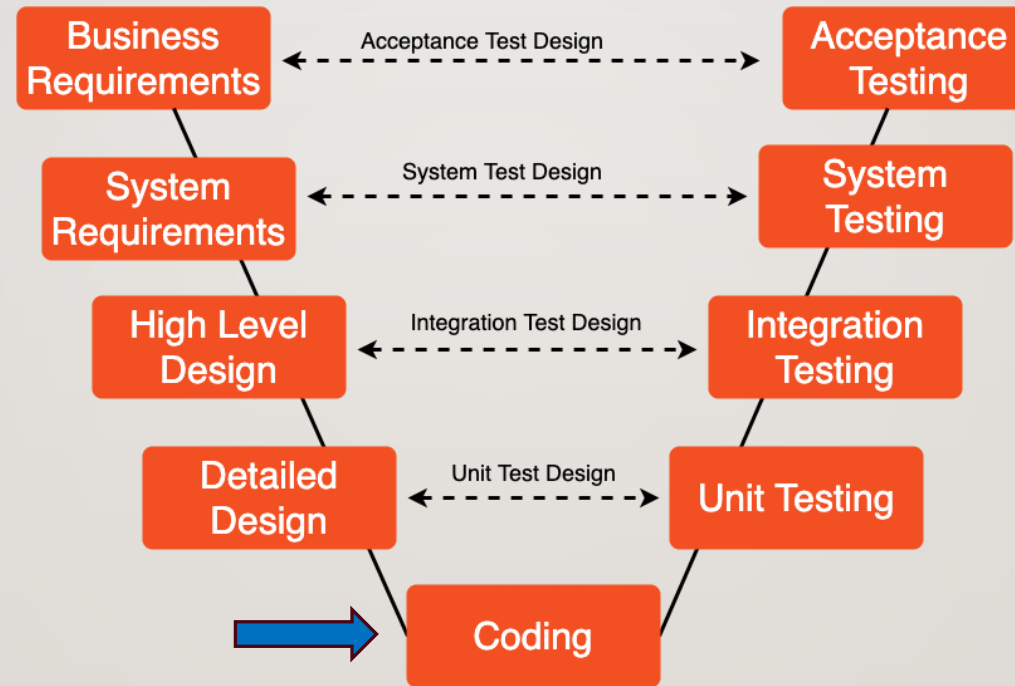
Perform Detail Design of AMR Controller Module using Process Flow Diagram

# DETAIL DESIGN REVIEW BY EACH TEAM

---

Using the process flow diagram present team's design

# SPRINT 2 – AMR CONTROLLER



SDLC - V Model - notepub.io

# CODING HINT

---

- Initial Pose
  - nav2
  - rviz2 – 2D estimate pose
  - ros2 topic echo /initialpose
- Sending Goals
  - nav2
  - Rviz2 – send goals
    - ActionClient
    - NavigateToPose
  - ros2 topic echo /amcl\_pose
- Stopping Navigation
  - NavigateToPose.cancel\_all\_goals\_async()
- Sending multiple goals
  - ActionClient
  - /follow\_waypoints
- Manual control of AMR Odometry
  - How to move forward, backward, left and right???

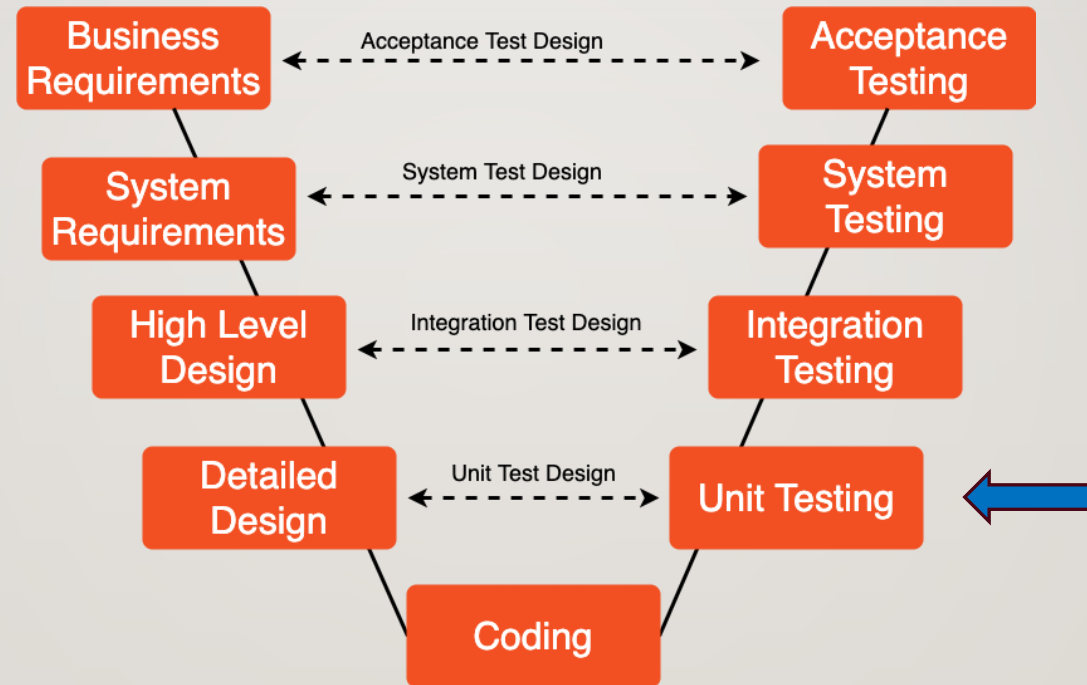


# EXPECTED OUTCOME

---

AMR navigates to avoid obstacles, ignores dummies, track, and follow target

# SPRINT 2 – AMR CONTROLLER



SDLC - V Model - notepub.io

# TEAM EXERCISE 7

---

Perform coding and testing of AMR Controller Module

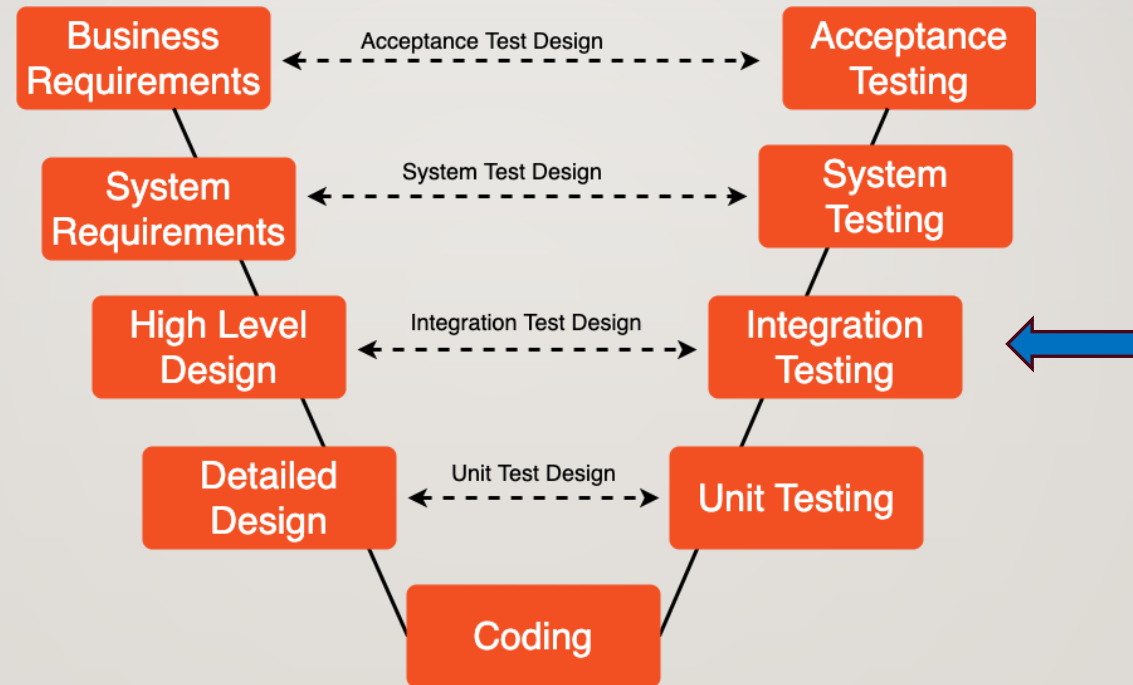
# RESULTS & CODE REVIEW BY EACH TEAM

---

Show actual results against the expected results and explain the code generated



# SPRINT 1&2 – DETECTION ALERT/AMR CONTROLLER INTEGRATION & TEST



SDLC - V Model - notepub.io

# EXPECTED OUTCOME

---

- Detection Alert and AMR Controller able to pass topics for necessary actions between

# TEAM EXERCISE 8

---

Perform integrate and test of Detection Alert and AMR Controller Modules

# RESULTS & CODE REVIEW BY EACH TEAM

---

Show actual results against the expected results and explain the code written



# PROJECT SPRINTS

---

- Detection Alert

- Camera Capture
- Object Detection
- Send messages to other subsystems

- AMR Controller

- Receive messages and act accordingly
- Move using (SLAM) with Obstruction avoidance
- Target Acquisition (Obj. Det.) and Tracking
- Follow target using camera and motor control

- System Monitor

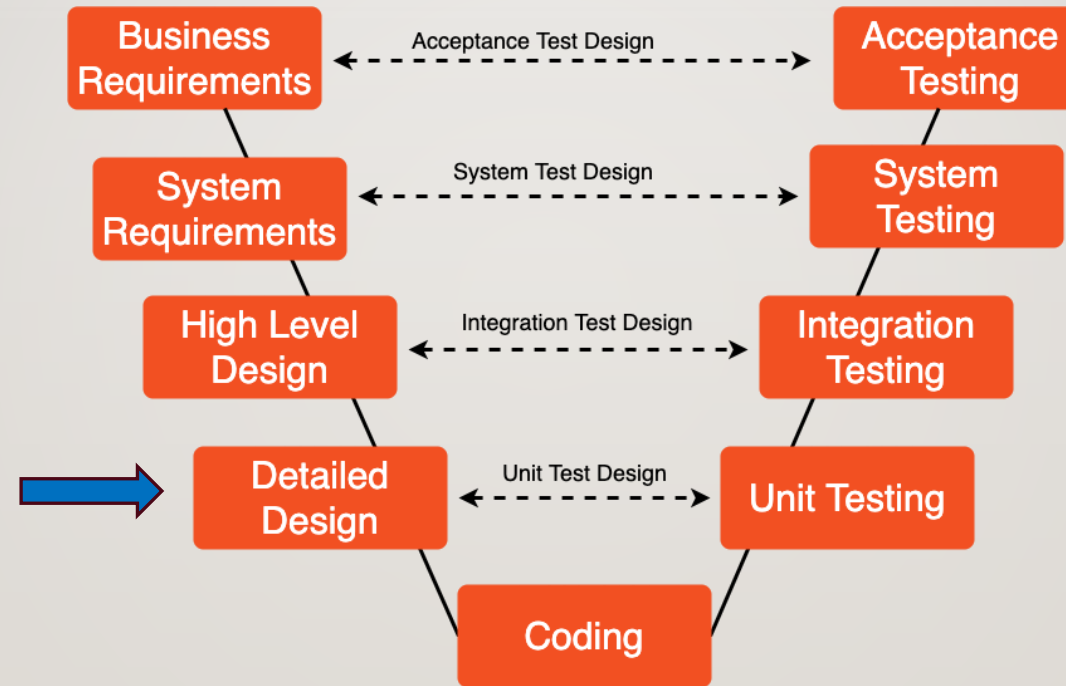
- Receive and Display Detection Camera and info
- Receive and Display AMR Camera and info
- Store, display, and report Information and Alerts

# SYSTEM MONITOR SPRINT

---



# SPRINT 3 – SYSTEM MONITOR



SDLC - V Model - notepub.io

# TEAM EXERCISE 9

---

Perform Detail Design of System Monitor Module using Process Flow Diagram

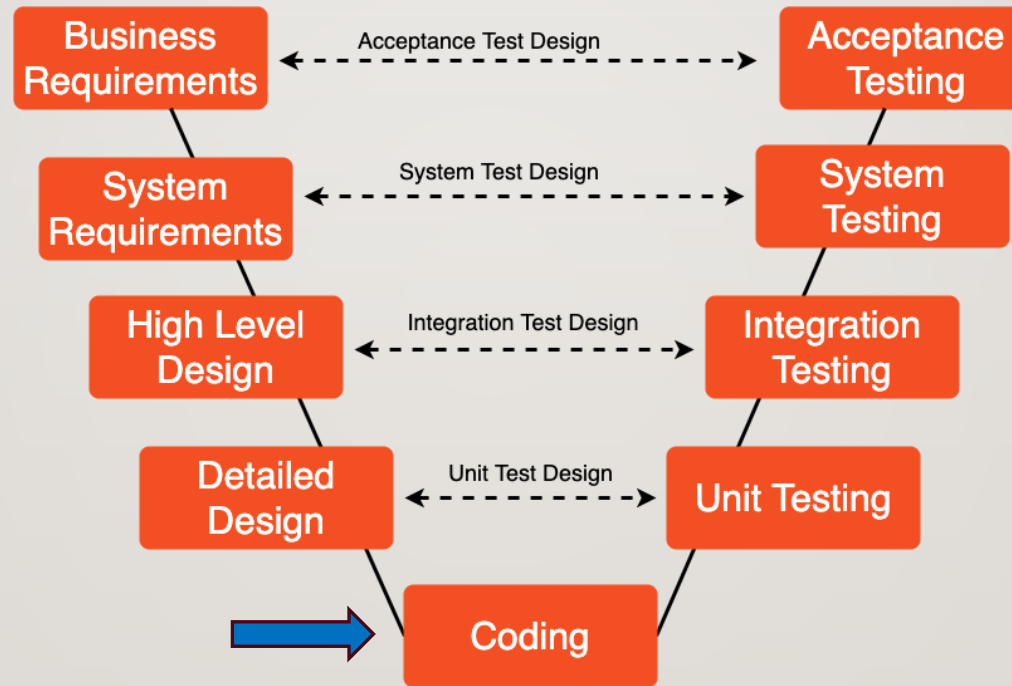
# DETAIL DESIGN REVIEW BY EACH TEAM

---

Using the process flow diagram present team's design



# SPRINT 3 – SYSTEM MONITOR



SDLC - V Model - notepub.io

# CODING HINTS

---

- Flask Basic Review
- SQLite Basic Review
- Webpage
  - Login page
  - Two video window
  - Alert Report
    - Status Captured and Following
- Database – SQLite
  - Login Data
  - Status Data

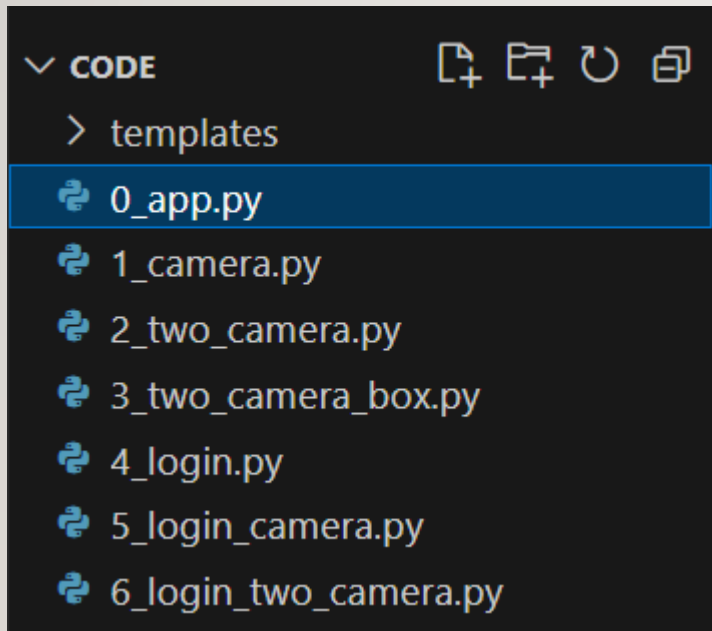
# INTRODUCTION TO FLASK

---

- What is Flask?  
A lightweight web framework for Python.
- Why Flask? Simple, flexible, good for beginners and small projects.
- `pip install Flask`
- `<project>/`
  - `|─ app.py` # Main Flask application file
  - `└─ templates/` # Folder for HTML templates
    - `└─── index.html`

# FLASK HINTS

---



- HTML Reference:

[HTML elements reference - HTML: HyperText Markup Language | MDN](https://developer.mozilla.org/en-US/docs/Web/HTML/Element)

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

- CSS

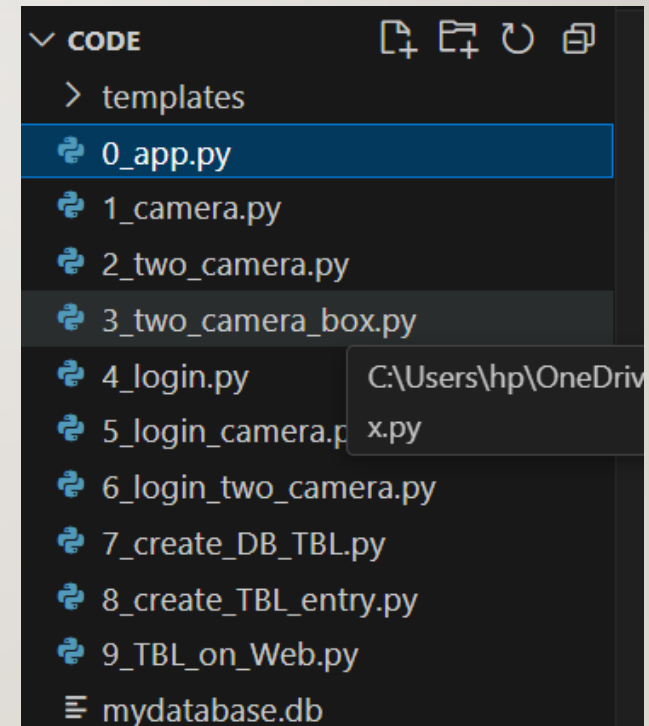
[CSS: Cascading Style Sheets | MDN](https://developer.mozilla.org/en-US/docs/Web/CSS)

<https://developer.mozilla.org/en-US/docs/Web/CSS>

# CODING HINTS

---

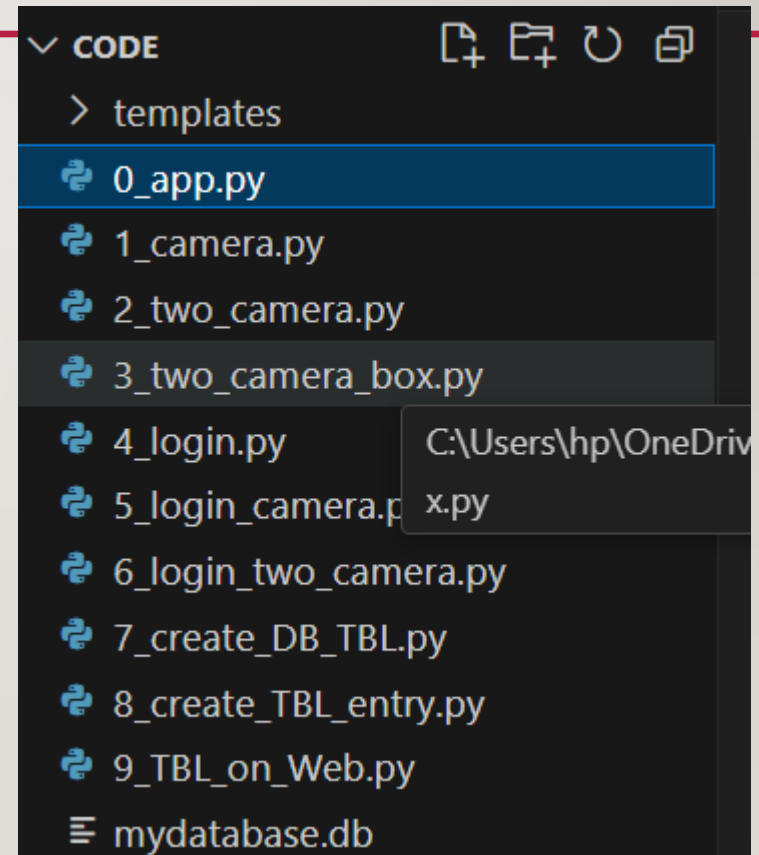
- Flask Basic Review
- SQLite Basic Review
  - SQLite is a lightweight, self-contained, serverless SQL database engine.





# CODING HINTS

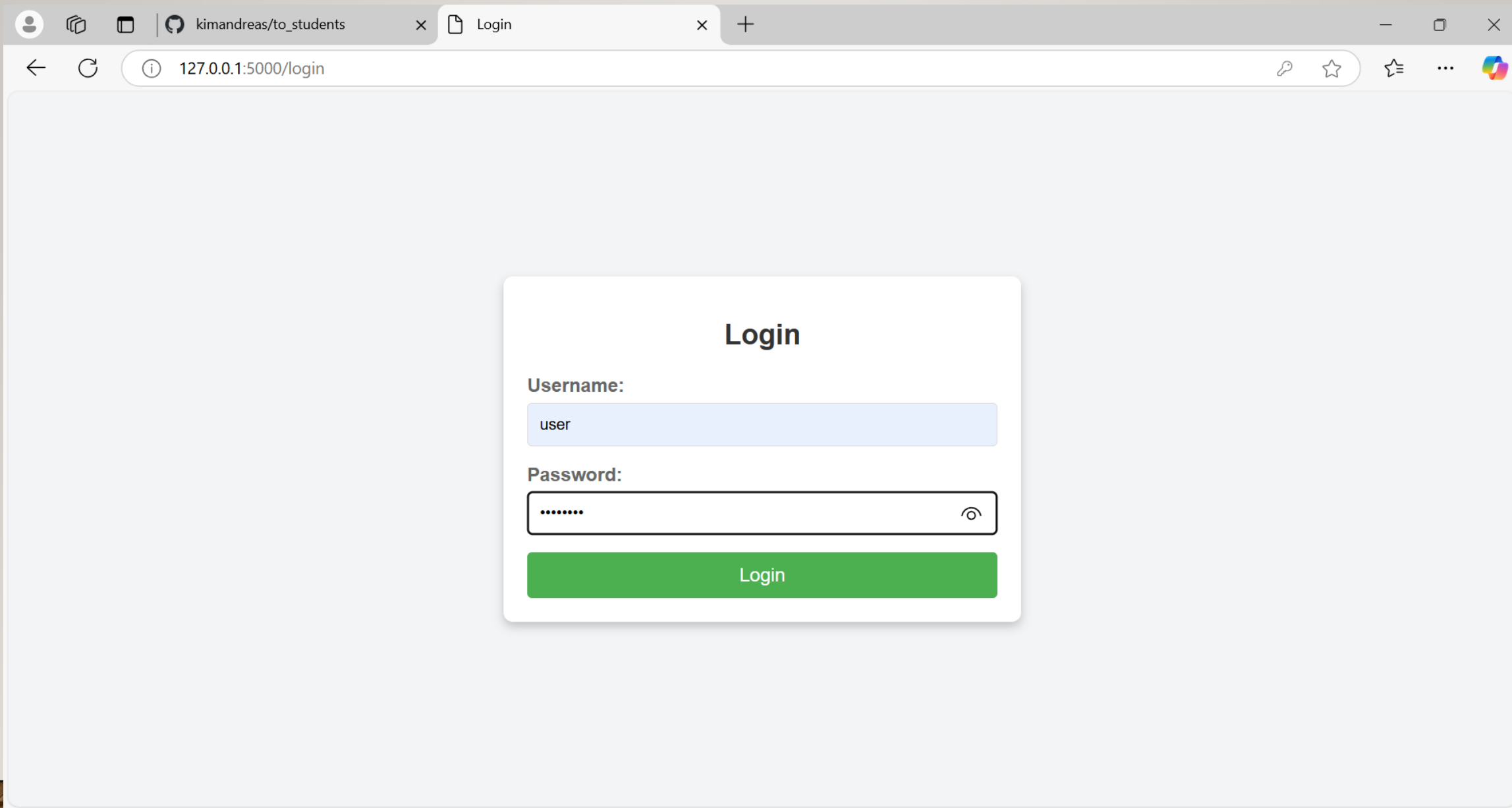
- Flask Basic Review
- SQLite Basic Review
- Webpage
  - Login page
  - Two video window
  - Alert Report
    - Status Captured and Following
- Database – SQLite
  - Detection Alert Data



# CODING HINTS

---

- Database – SQLite viewer
  - `sudo apt install sqlitebrowser`
  - VSCode sqlite viewer extension



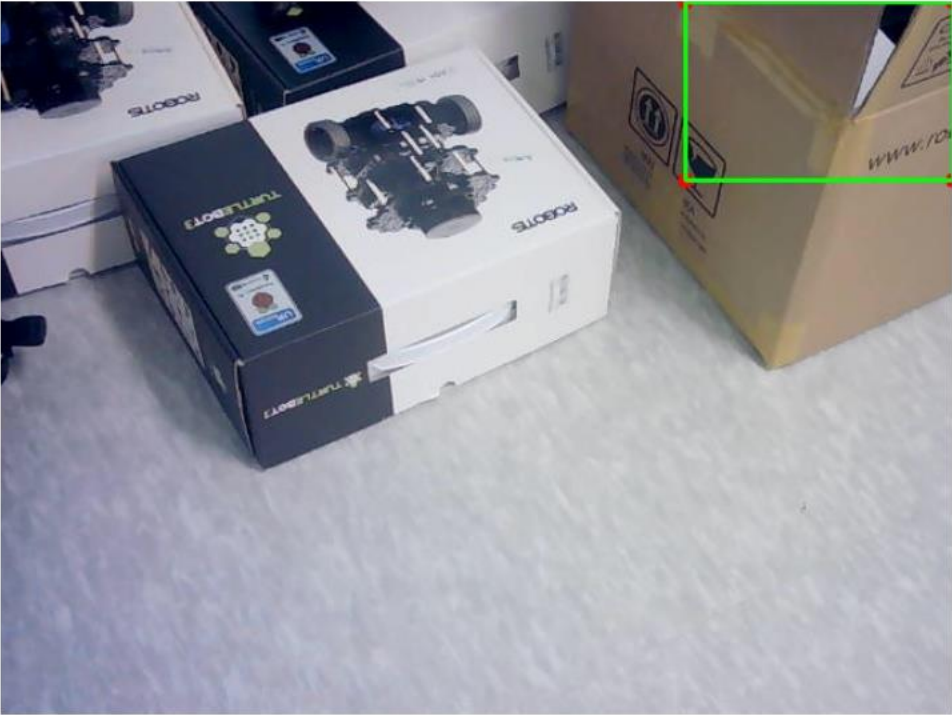
kimandreas/to\_students

Welcome

127.0.0.1:5000/welcome


welcome, user!

You are now logged in.



Violations Detected

ID	Name	Date & Time
0	Truck	2024-11-06 10:30:22



Track and Following

ID	Name	Date & Time
1	Dummy	2024-11-06 10:30:22

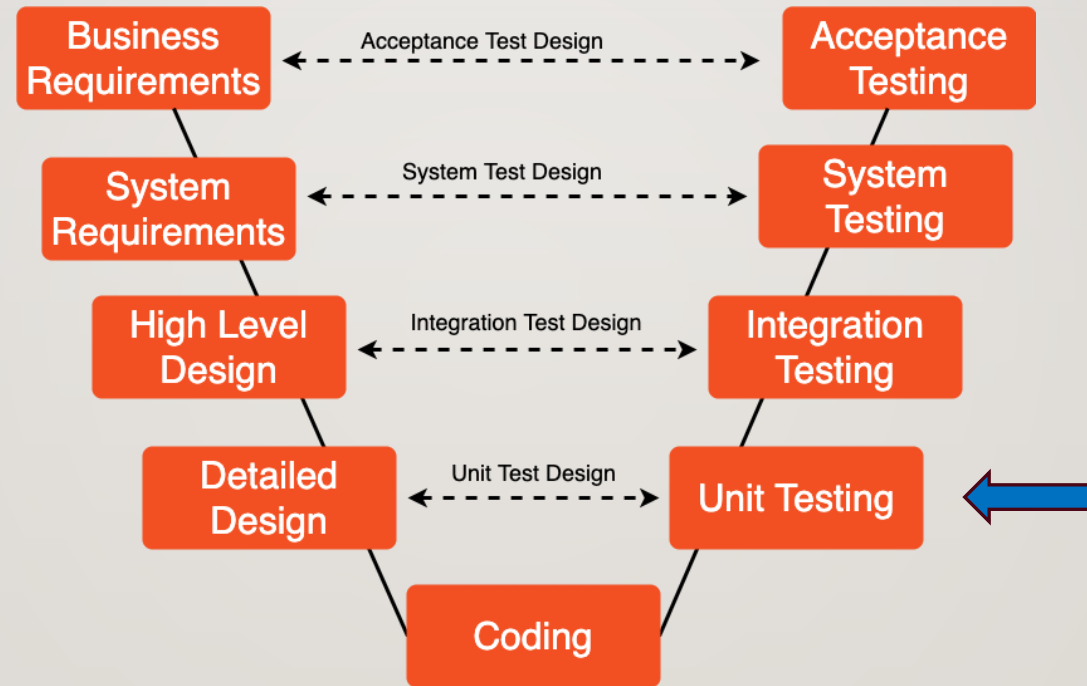
# EXPECTED OUTCOME

---

- Sysmon with two windows and related Detection and tracking info
- ROS Nodes, Services, Topics



# SPRINT 3 – SYSTEM MONITOR



SDLC - V Model - notepub.io

# TEAM EXERCISE 10

---

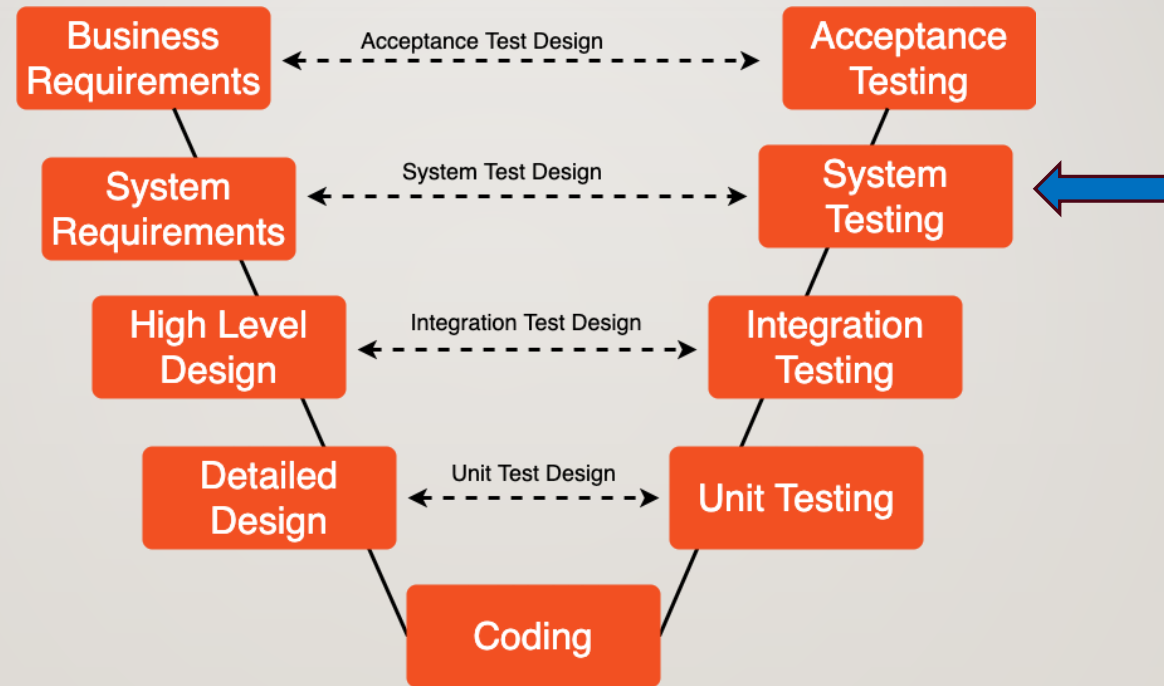
Perform coding and testing of System Monitor Module

# RESULTS & CODE REVIEW BY EACH TEAM

---

Show actual results against the expected results and explain the code written

# SYSTEM INTEGRATION & TEST



SDLC - V Model - notepub.io

# THE LAST DAY

---

- 9:30 – 4:00 p.m
  - System Integration & Test
  - Final Presentation Prep
- 4:00 – 5:40 p.m.
  - Live Demonstration 5 minutes
  - Presentation 15 minutes
- Equipment Return and Rap up



# 최종 프로젝트 발표

---



# FINAL PRESENTATION MATERIAL PLANNING

---

- Solution Overview
  - Key Issues and Challenges
    - How did you overcome
  - Required Solution Improvements
  - Lessons Learned
  - Team Contribution
- 20 minutes

# 팀원 과 업무 책임

---



- 업무 책임
  - ...
- 숙련된 기술
  - ....

새해복 많이받으세요!!

---

