GOOD MORNING!
早上好!
안녕하세요!

PROJECT INTRODUCTION

DAY 1 (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

- 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 3

- 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 & System Monitor

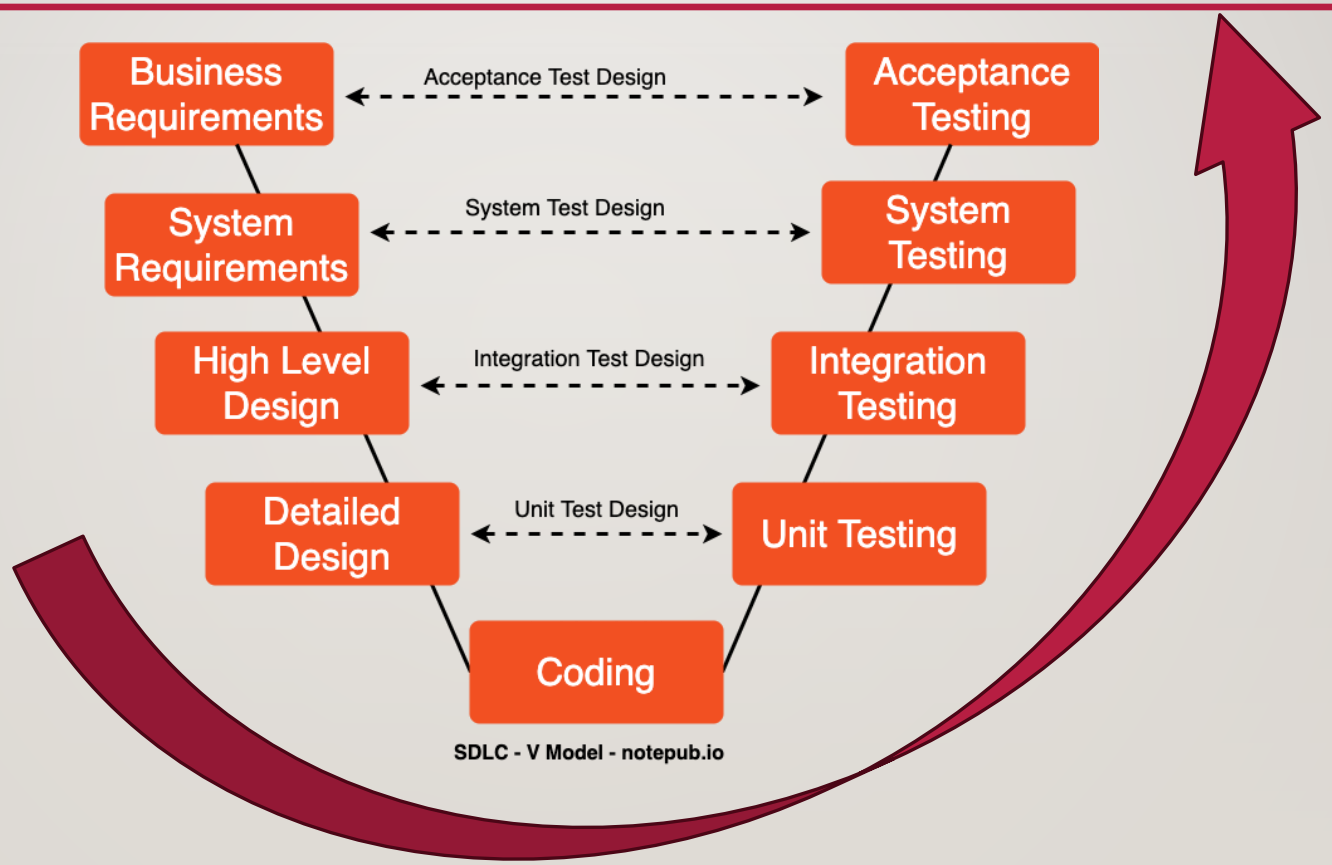
DAY 4

- 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

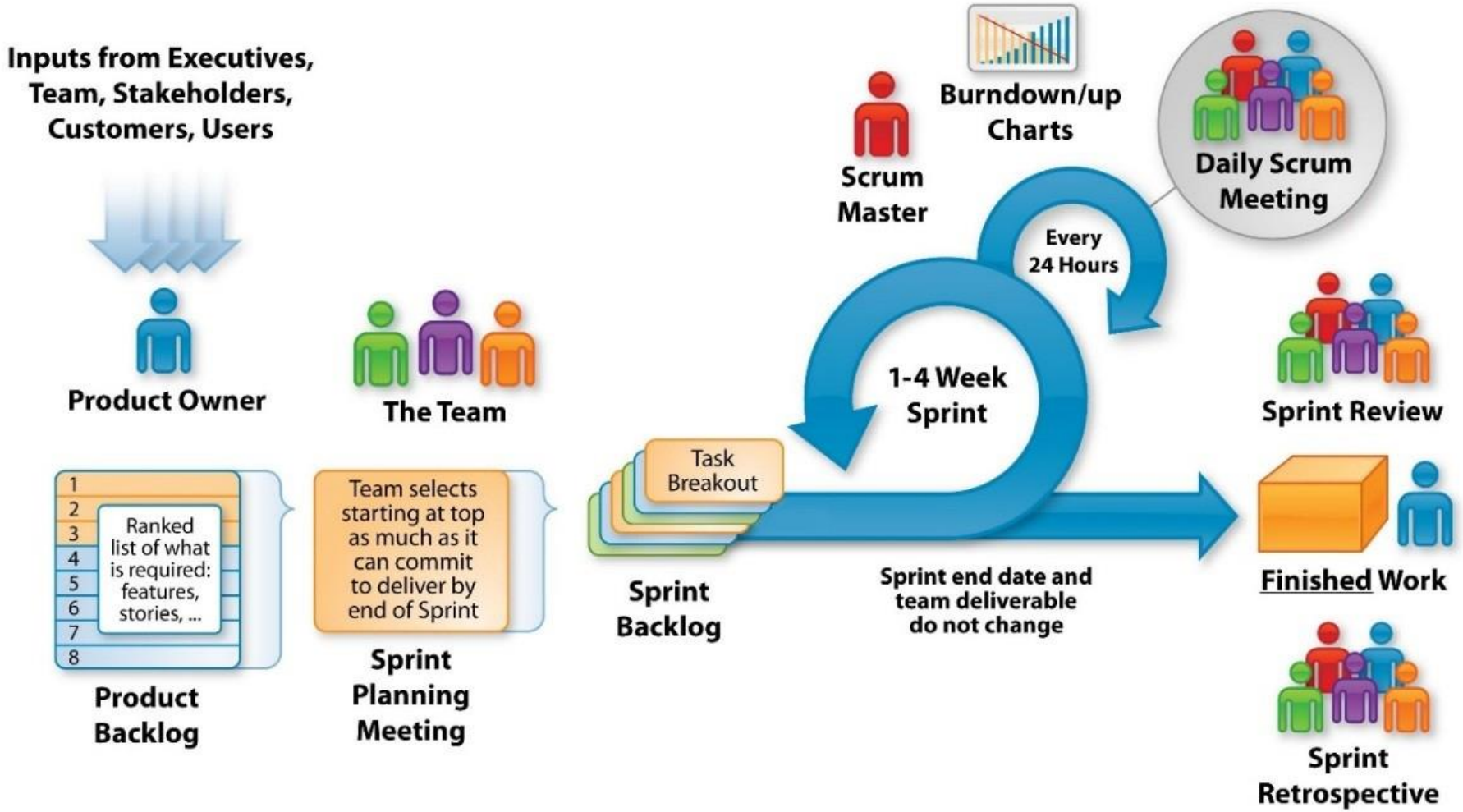
프로젝트 RULE NUMBER ONE!!!

Have Fun Fun Fun!

SW DEVELOPMENT PROCESS



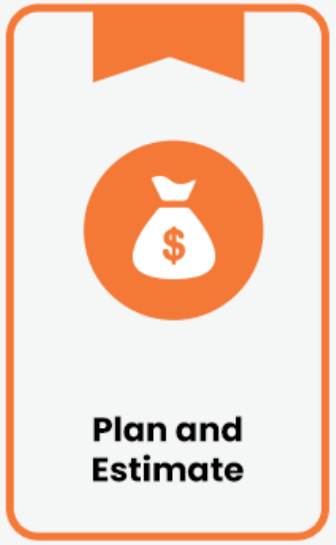
The Agile - Scrum Framework



5 Stages of Scrum Sprint

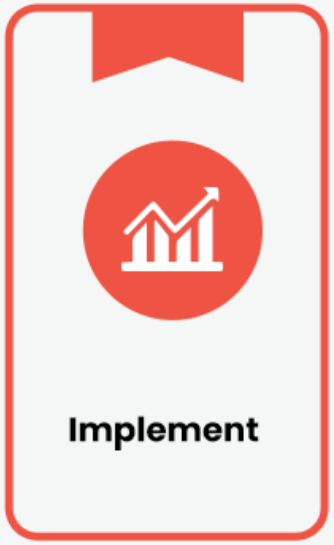
Initiate

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.



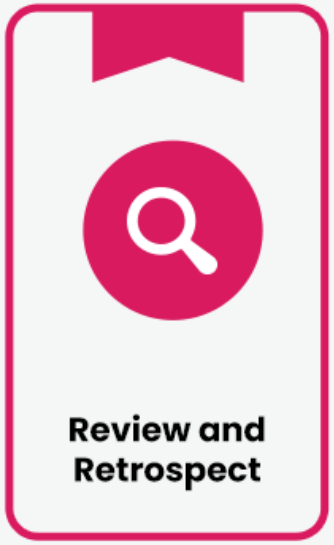
Plan and Estimate

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.



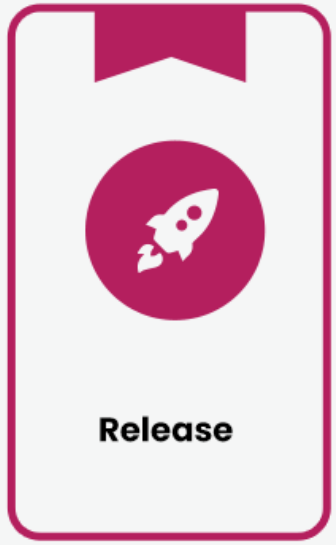
Implement

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.



Review and Retrospect

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.

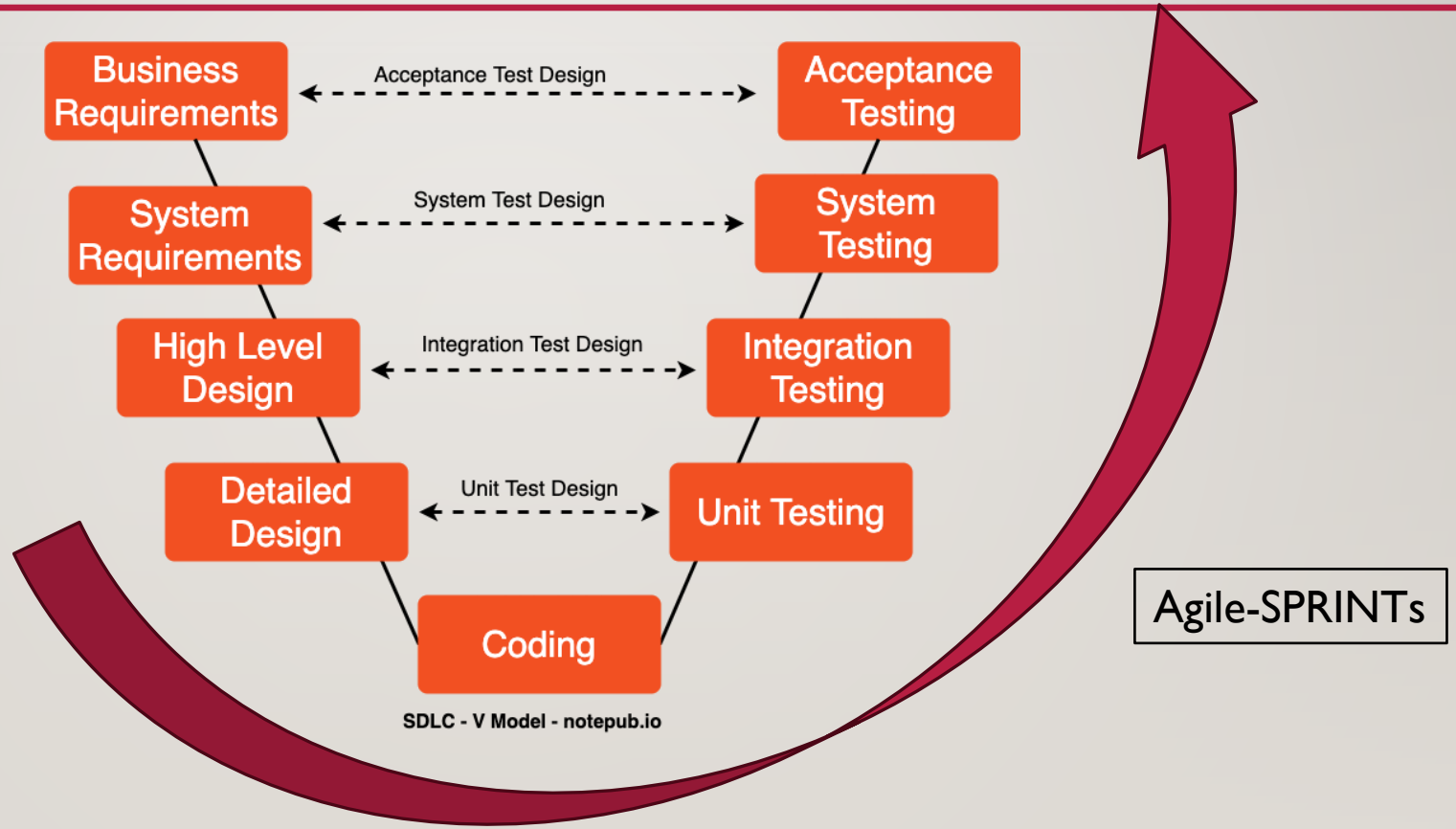


Release

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

maruti techlabs

SW DEVELOPMENT PROCESS



Agile-SPRINTs

TEAM EXERCISE 6

Perform coding and testing of Detection Alert Module

RESULTS & CODE REVIEW BY EACH TEAM

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

PROJECT SPRINTS

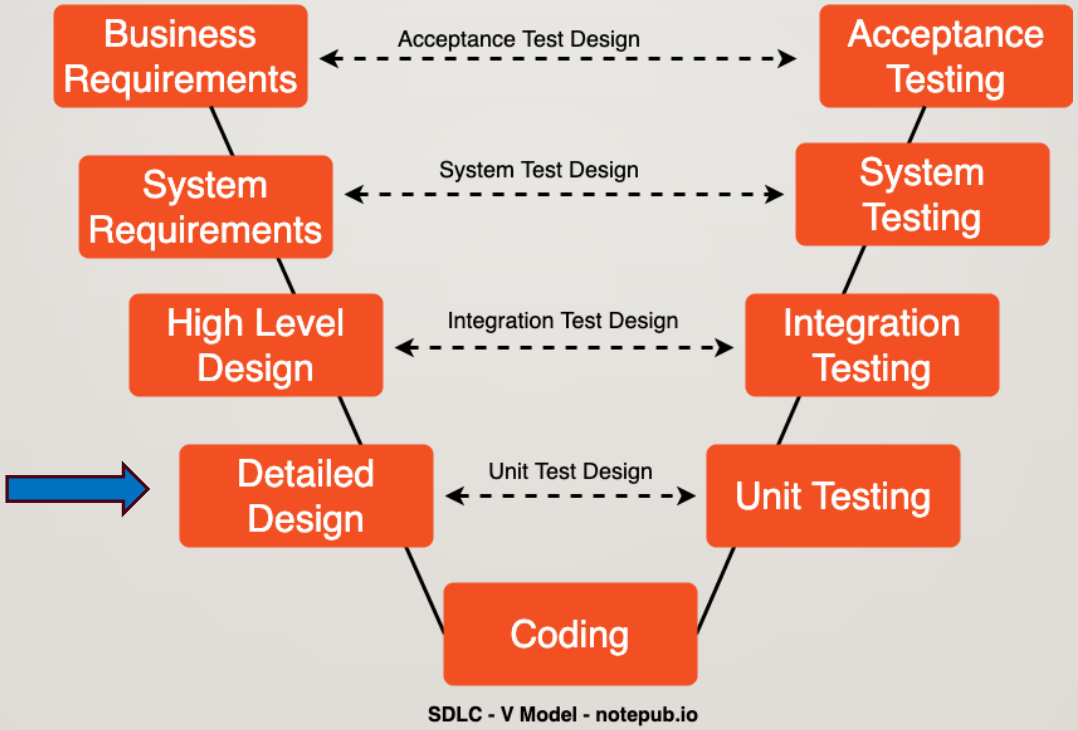
- Detection Alert
  - Camera Capture
  - Object Detection
  - Send messages to other subsystems

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

- 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 SPRINT

SPRINT 2 – SYSTEM MONITOR



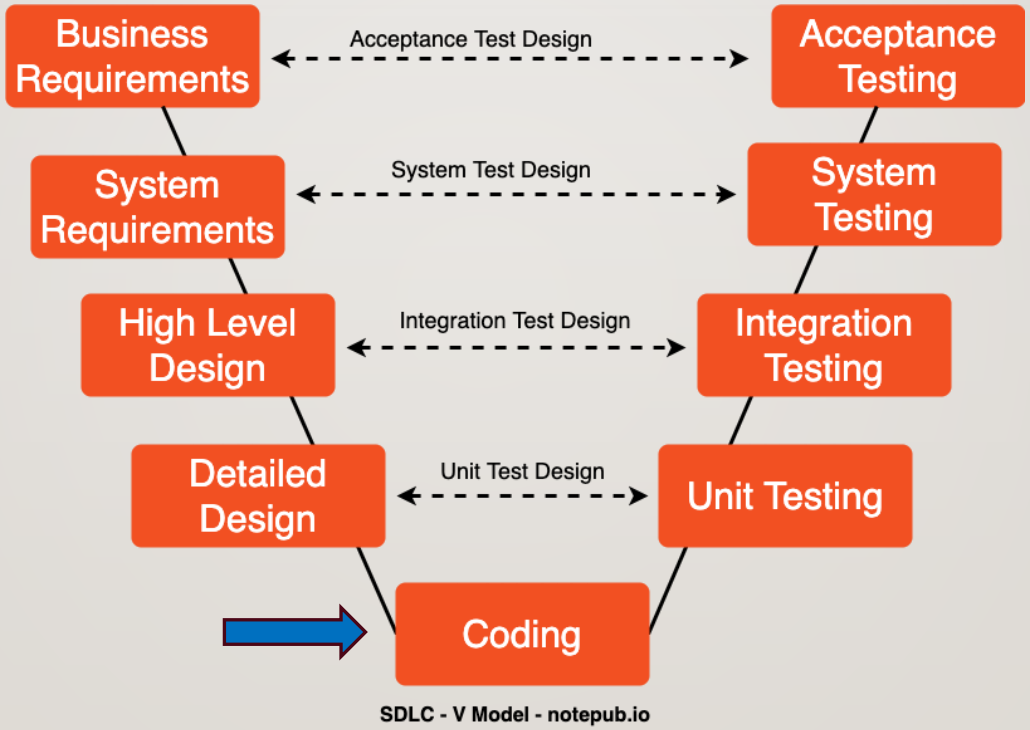
TEAM EXERCISE 7

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 2 – SYSTEM MONITOR



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

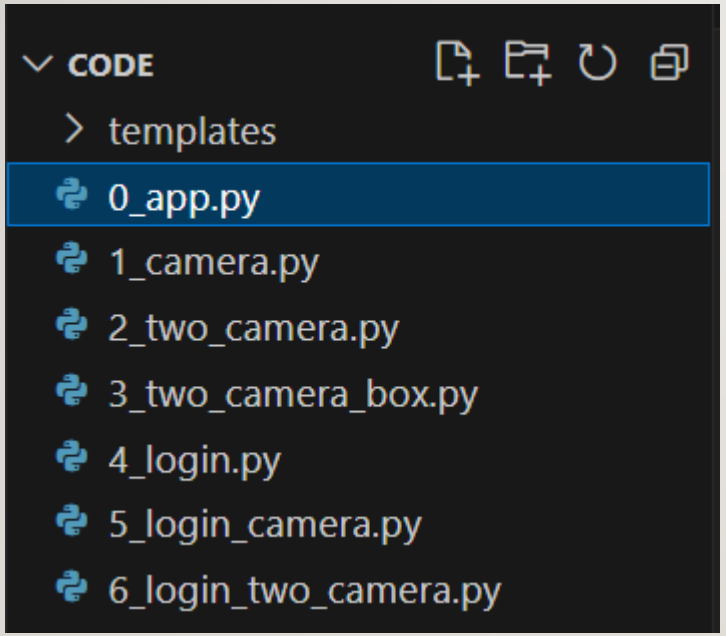
- What is Flask?

  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

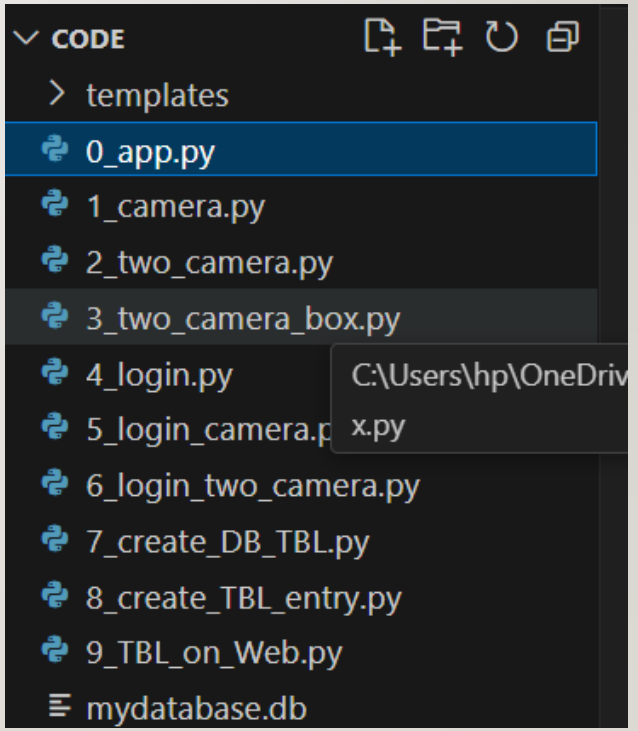
- CSS

  CSS: Cascading Style Sheets | MDN

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

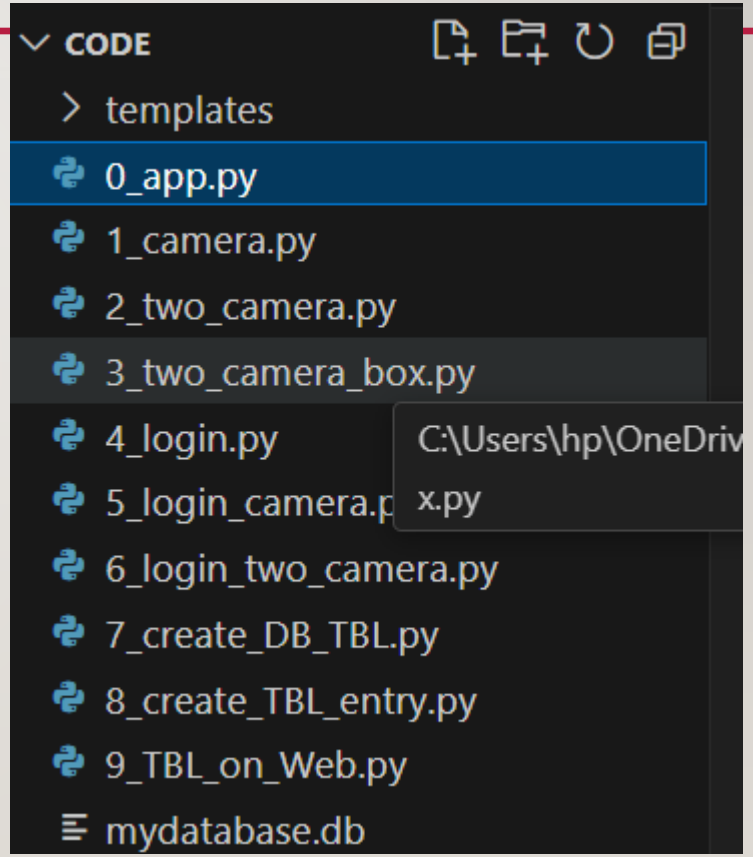
CODING HINTS

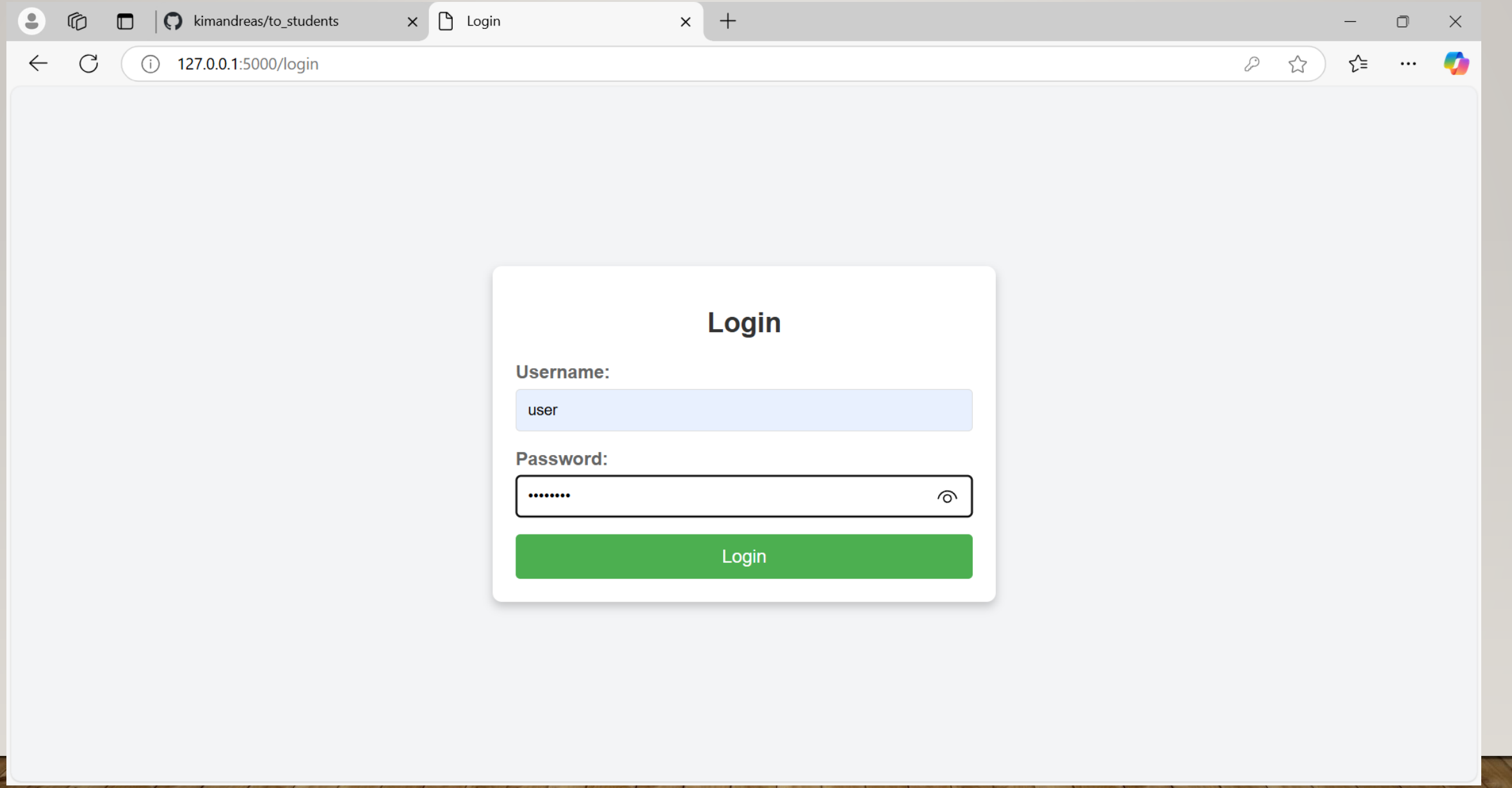
- Flask Basic Review
- SQLite Basic Review
  - SQLite is a lightweight, self-contained, serverless SQL database engine.



CODING HINTS

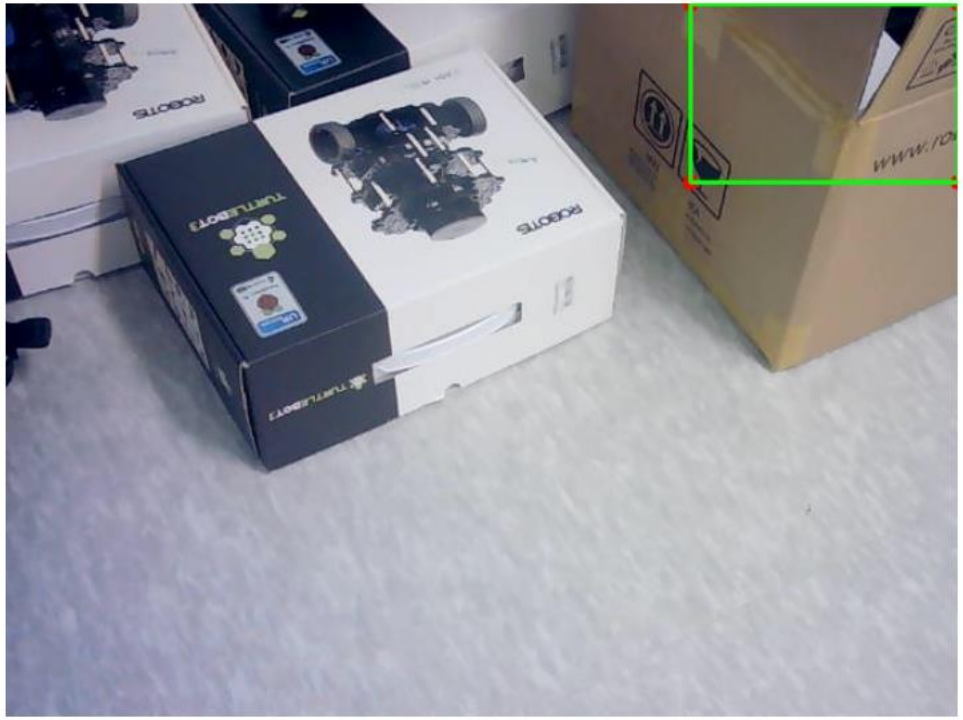
- Flask Basic Review
- SQLite Basic Review
- Webpage
  - Login page
  - Two video window
  - Alert Report
    - Status Captured and Following
- Database – SQLite
  - Detection Alert Data




kimandreas/to_students
Login
127.0.0.1:5000/login
Login
Username:
user
Password:
Login

Welcome, user!

You are now logged in.



Violations Detected

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

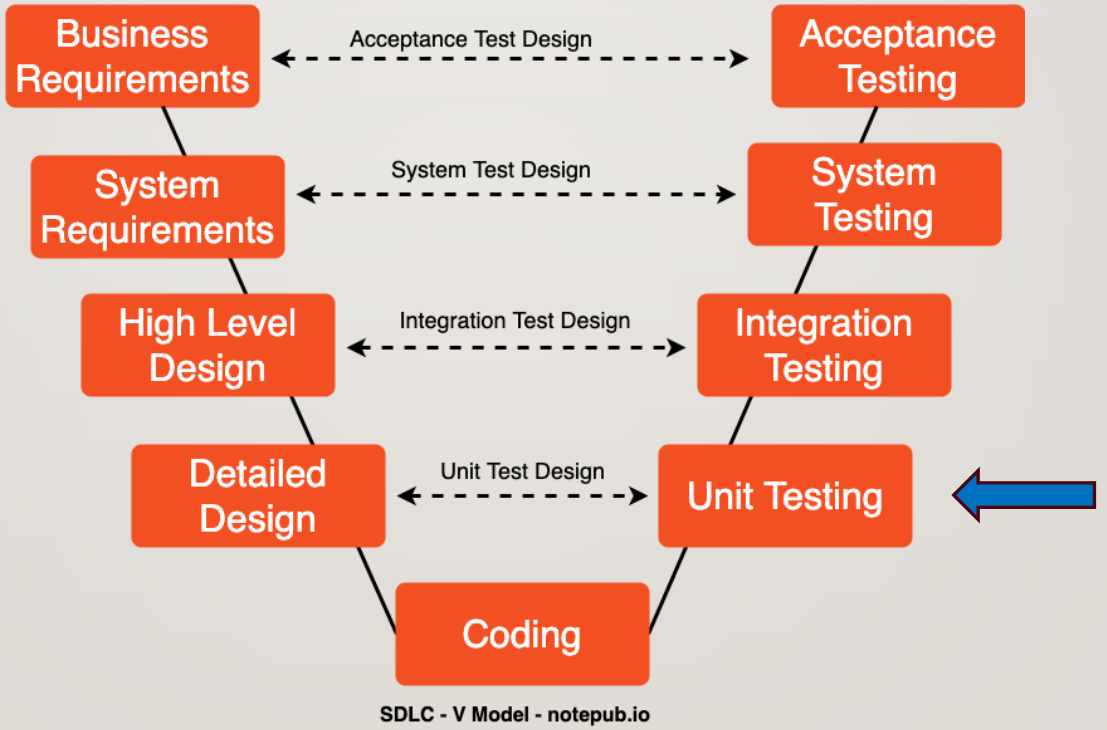
Track and Following

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 2 – SYSTEM MONITOR



TEAM EXERCISE 8

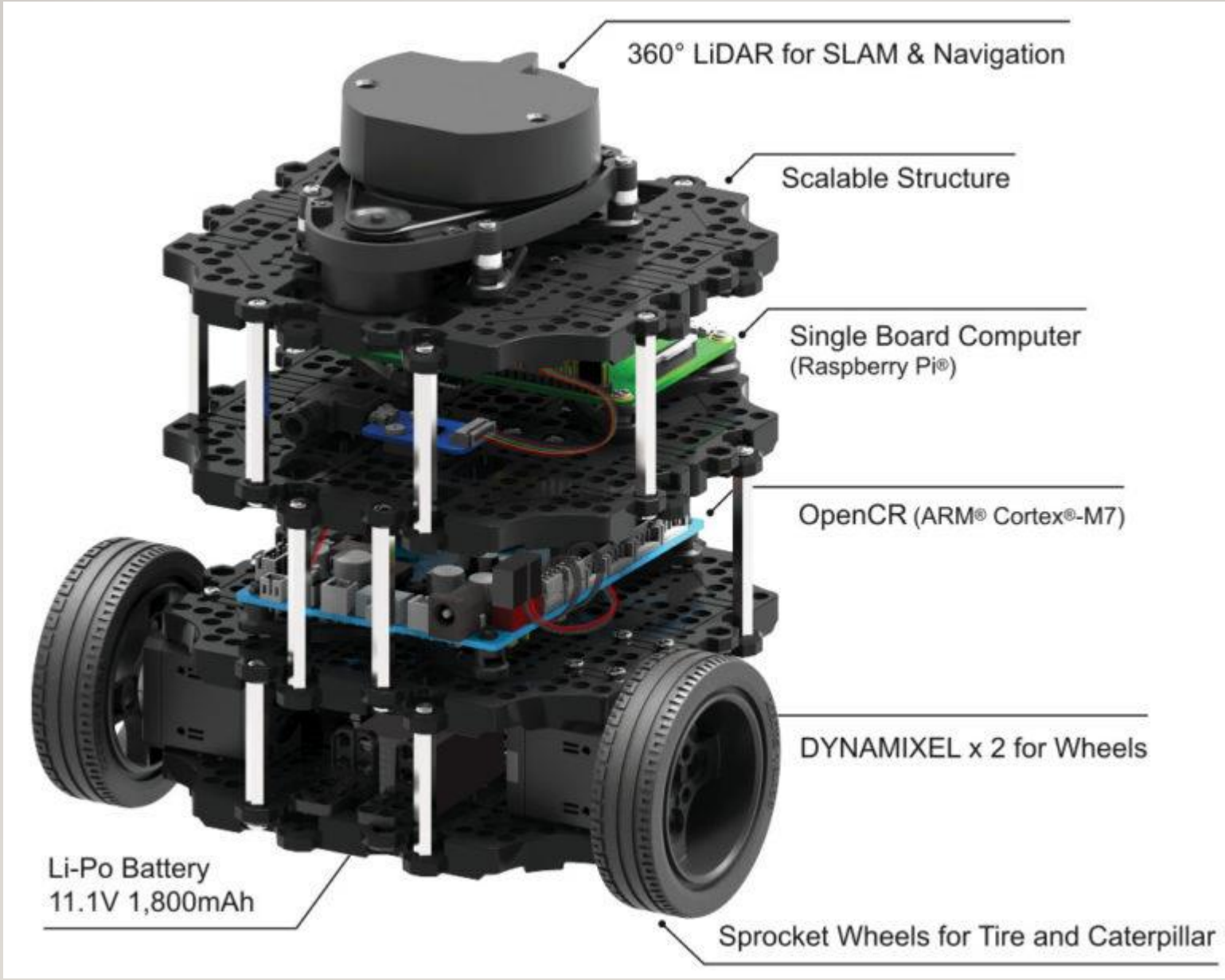
Perform coding and testing of System Monitor Module

RESULTS & CODE REVIEW BY EACH TEAM

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

INTRODUCTION TO AMR

- TurtleBot3
- https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/


360° LiDAR for SLAM & Navigation
Scalable Structure
Single Board Computer
(Raspberry Pi®)
OpenCR (ARM® Cortex®-M7)
DYNAMIXEL x 2 for Wheels
Li-Po Battery
11.1V 1,800mAh
Sprocket Wheels for Tire and Caterpillar

SETUP PC FOR AMR

TURTLEBOT3

https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/

CHECK IF INSTALLED

$ apt list | grep gazebo

$ apt list | grep carto

INSTALL DEPENDENT ROS 2 PACKAGES

If not already installed,

$ sudo apt install ros-humble-gazebo-*

$ sudo apt install ros-humble-cartographer

$ sudo apt install ros-humble-cartographer-ros

SETUP PC FOR AMR

INSTALL DEPENDENT ROS 2 PACKAGES

$ Check if installed

    $ apt list | grep gazebo

    $ apt list | grep carto

If not,

$ sudo apt install ros-humble-gazebo-*

$ sudo apt install ros-humble-cartographer

$ sudo apt install ros-humble-cartographer-ros

TURTLEBOT3

https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/

SETUP PC FOR AMR

CHECK IF ALREADY INSTALLED

$ apt list | grep dynam

$ apt list | grep turtlebot3

INSTALL TURTLEBOT3 PACKAGES

If not already installed,

$ source ~/.bashrc

$ sudo apt install ros-humble-dynamixel-sdk

$ sudo apt install ros-humble-turtlebot3-msgs

$ sudo apt install ros-humble-turtlebot3

SETUP PC FOR AMR

- If you want to download the source code

$ mkdir -p ~/turtlebot3_ws/src

$ cd ~/turtlebot3_ws/src/

$ git clone -b humble-devel https://github.com/ROBOTIS-GIT/DynamixelSDK.git

$ git clone -b humble-devel https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git

$ git clone -b humble-devel https://github.com/ROBOTIS-GIT/turtlebot3.git

$ cd ~/turtlebot3_ws

$ colcon build --symlink-install

$ echo 'source ~/turtlebot3_ws/install/setup.bash' >> ~/.bashrc

$ source ~/.bashrc

SETUP ROS ID

PC

```
$ echo 'export ROS_DOMAIN_ID=1
  #TURTLEBOT3' >> ~/.bashrc
$ source ~/.bashrc
$ env | grep ROS
```

- 1,2,3,4,5

HOW TO CONNECT TO AMR

PC

$ sudo ufw disable

- disables firewall for ubuntu systems

HOW TO CONNECT TO AMR

PC/AMR

Connect to rokey or rokey_APTest

Must be same network for both

*rokey_APTest does not have internet access

AMR

- Get AMR IP address by physically connect by monitor and keyboard

$ Ifconfig

```
Wireless LAN adapter Wi-Fi 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::2bd0:50e1:9694:44%13
   IPv4 Address. . . . . . . . . . . : 192.168.10.14
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.10.1
```

You can find then AMR ID rokey<n> in the linux prompt

HOW TO CONNECT TO AMR

PC TERM1

$ dpkg -l | grep openssh

If not installed…

$ sudo apt install openssh-server –y

$ ssh -X rokey<n>@<ip_address>

*allows Vscode/Rviz to run

HOW TO CONNECT TO AMR

PC TERM1

$ ros2 run demo_node_cpp talker

SSH AMR TERM1

$ ros2 run demo_node_cpp listener

CREATE YOUR WORKSPACE UNDER AMR $HOME DIRECTORY

AMR

$ mkdir

~/rokey1_<grp_letter><grp_num>_ws

(i.e. mkdir ~/rokey1_A2_ws)

- Put all your file under this directory and remove at the end of the class
- Delete the directory at the end of the class

SETTING UP TO USE TURTLEBOT3

PC TERM 1

$ ssh -X
rokey<n>@<ip_address>

SSH AMR TERM 1

$

SETTING UP TO USE TURTLEBOT3

SSH AMR TERM 1

$ cat ~/.bashrc

Check if export command on the right is in the .bashrd, if not execute the command on the right

SSH AMR TERM 1

$ echo ‘export TURTLEBOT3_MODEL=burger’ >> ~/.bashrc

$ echo 'export ROS_DOMAIN_ID=1 #TURTLEBOT3' >> ~/.bashrc

(ID = 1,2,3,4,5 depends on your team number

SETTING UP TO USE TURTLEBOT3

PC TERM 1

$ cat ~/.bashrc

Check if export command on the right is in the .bashrd, if not execute the command on the right

PC TERM 1

$ echo ‘export TURTLEBOT3_MODEL=burger’ >> ~/.bashrc

$ echo 'export ROS_DOMAIN_ID=1 #TURTLEBOT3' >> ~/.bashrc

(ID = 1,2,3,4,5 depends on your team number)

USING AMR TELEOP

SSH AMR TERM 1

$ source ~/.bashrc

$ ros2 launch turtlebot3_bringup robot.launch.py

….. running

PC TERM2

$ source ~/.bashrc

$ ros2 run turtlebot3_teleop teleop_keyboard

USING AMR TELEOP

PC TERM1

$ ssh -X rokey<n>@<ip_address>

SSH AMR TERM 1

$ ros2 launch turtlebot3_bringup robot.launch.py

…..running

PC TERM2

$ ros2 run turtlebot3_teleop teleop_keyboard

…..running

DIGITAL MAPPING

STEP1 : SSH AMR TERM 1

```
$ source ~/bashrc
$ ros2 launch turtlebot3_bringup
  robot.launch.py
```

STEP2: SSH AMR TERM 2

```
$ source ~/bashrc
$ ros2 launch turtlebot3_cartographer
  cartographer.launch.py
```

DIGITAL MAPPING

STEP1: SSH AMR TERM 1

```
$ ros2 launch turtlebot3_bringup
  robot.launch.py

….. running
```

STEP2: SSH AMR TERM 2

```
$ ros2 launch turtlebot3_cartographer
  cartographer.launch.py

….. running
```

STEP3: PC TERM 2

```
$ source ~/bashrc

$ ros2 run turtlebot3_teleop
  teleop_keyboard
```

DIGITAL MAPPING

STEP1: SSH AMR TERM 1

```
$ ros2 launch turtlebot3_bringup
  robot.launch.py

….. running
```

STEP2: SSH AMR TERM 2

```
$ ros2 launch turtlebot3_cartographer
  cartographer.launch.py

….. running
```

STEP3: PC TERM 2

```
$ ros2 run turtlebot3_teleop
  teleop_keyboard

….. running
```

STEP4: SSH AMR TERM 3(AT THE END)

```
$ source ~/bashrc

$ ros2 run nav2_map_server
  map_saver_cli -f ~/<my_dir>/map
```

NAVIGATION W/ MAP

STEP1 : SSH AMR TERM 1

```
$ source ~/bashrc
$ ros2 launch turtlebot3_bringup
  robot.launch.py
```

STEP2: SSH AMR TERM 2

```
$ source ~/.bashrc
$ ros2 launch turtlebot3_navigation2
  navigation2.launch.py
  map:=<map_file_path> (i.e:
  $HOME/my_dir/map/map.yaml)
```

NAVIGATION W/ MAP

STEP1 : SSH AMR TERM 1

```
$ ros2 launch turtlebot3_bringup
  robot.launch.py

….running
```

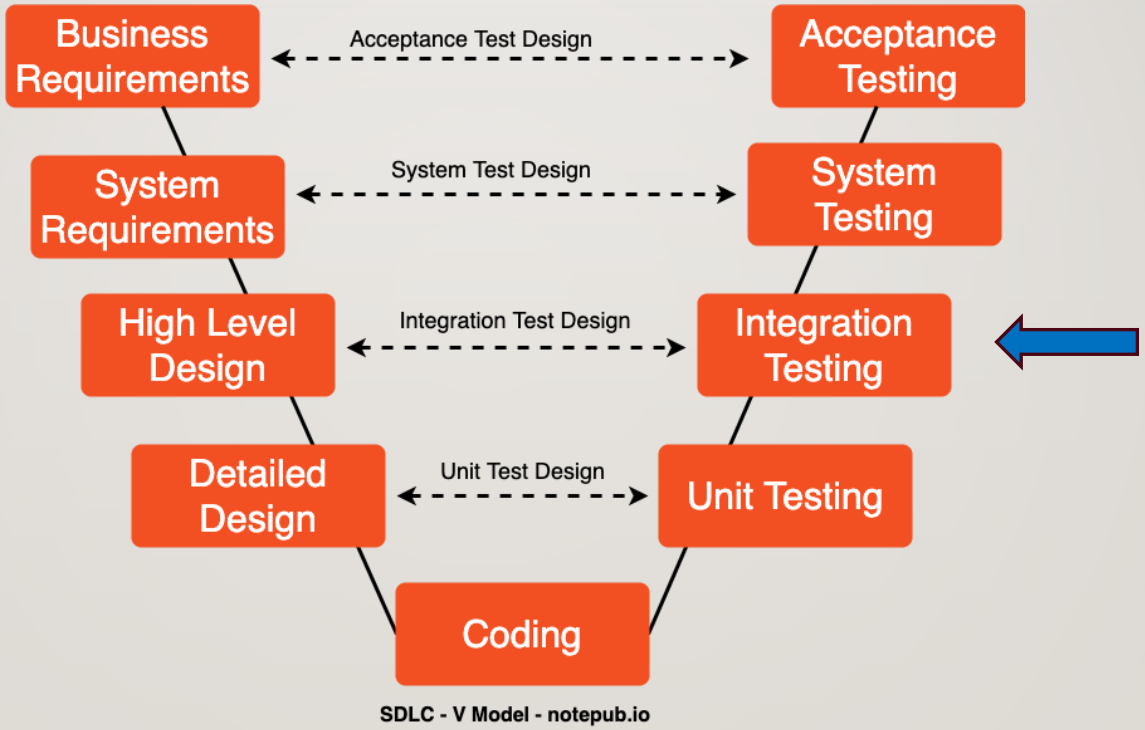
STEP2: SSH AMR TERM 2

```
$ ros2 launch turtlebot3_navigation2
  navigation2.launch.py
  map:=<map_file_path> (i.e:
  $HOME/my_dir/map/map.yaml)

….running
```

*Perform 2D Pose Estimate and Send Goal

SPRINT 1&2 – SYSTEM MONITOR/DETECTION ALERT INTEGRATION & TEST



EXPECTED OUTCOME

- Detection Alert and SysMon able to pass topics to update video and data

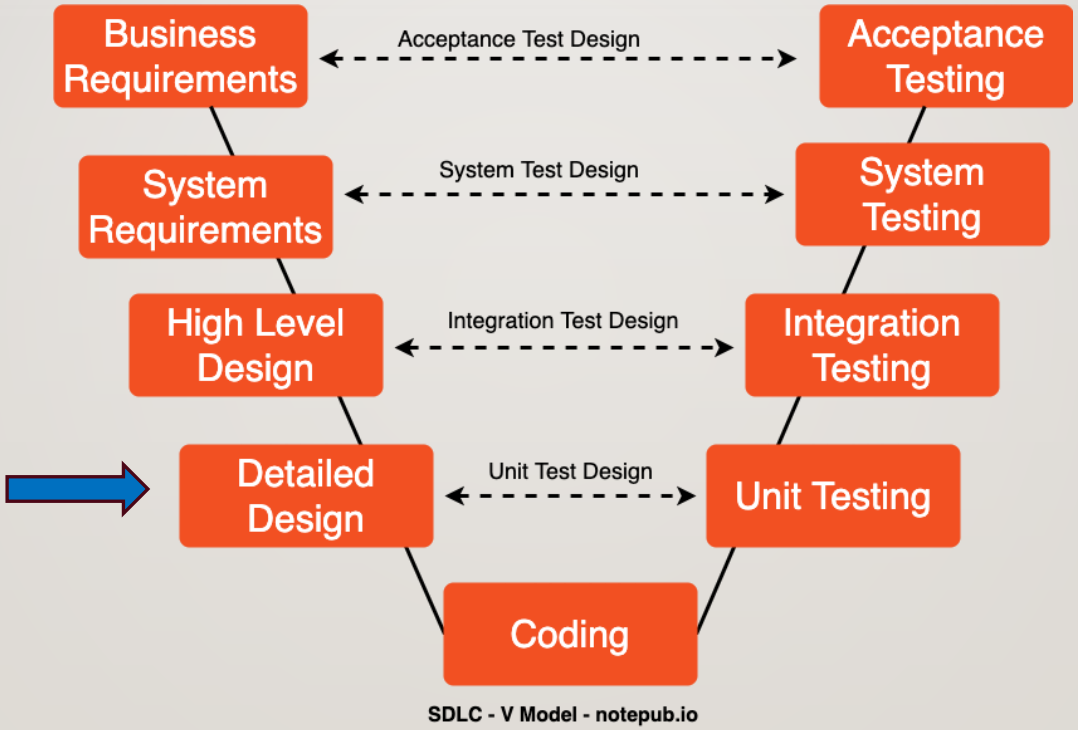
TEAM EXERCISE 9

Perform integrate and test of System Monitor and Detection Alert Modules

RESULTS & CODE REVIEW BY EACH TEAM

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

SPRINT 3 – AMR CONTROLLER



TEAM EXERCISE 10

Perform Detail Design of AMR Controller Module using Process Flow Diagram