

# GOOD MORNING!

早上好!

안녕하세요!

---

DAY 7

# 2 PROJECTS

---

- Mini Project (Individual Team)
  - For learning techniques

시스템 개발 프로세스의 이해 개발 환경 구축
-----------------------------

AI VISION 기술 탐색 및 검증
----------------------

로봇 AMR 제어 기술 탐색 및 검증
----------------------

웹 시스템 모니터 기술 탐색 및 검증
----------------------

- Final Project (2 Teams in One)

통합 시스템 설계 및 개발
----------------

시스템 발표 및 시연
-------------

# FINAL PROJECT DESCRIPTION

---



# DAY I

---

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



# DAY 2/3 (MINI PROJECT)

---

- Yolo객체 인식 모델 활용과 성능 평가 방법 이해
- Custom Dataset과 Fine Tuning으로 자체 객체 인식 모델 구현 및 평가
- (Optional)경량화 모델 등 개별 요구사항에 적합한 모델 탐색 및 성능 검증
- 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 3/4 (MINI PROJECT)

---

- AMR(Autonomous Mobile Robot)  
Turtlebot4 개발 환경 구축
- 로봇 개발 환경에 완성 모델 서빙 및 테스트 / 로봇 H/W, 제반 환경의 한계점 도출
  - Tracking 데이터 수집(bus, truck, tank 등)
  - Tracking 데이터 라벨링
  - YOLOv8 기반 학습
  - YOLOv8 Object **Tracking**
- Turtlebot4 시뮬레이션 환경 구축
  - SLAM과 Map 생성 및 파라미터 튜닝 (Localization, AMCL)
  - AutoSLAM으로 맵 생성

# DAY 5/6 (MINI PROJECT)

---

- Turtlebot4 API를 활용한 Initial Pose Navigate\_to Pose 구현
- Turtlebot4 API를 활용한 Navigate\_Through\_pose, Follow Waypoints 구현
- 로봇 개발 환경에 적용 및 테스트 / 로봇 H/W, 제반 환경의 한계점 도출
- AMR기반 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
- Integrate and test with Detection

# DAY 7 (FINAL PROJECT)

---

- 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 7 (FINAL PROJECT)

---

- 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 7 (FINAL PROJECT)

---

- 시스템 설계 및 프로세스 정립
- 비즈니스 요구 사항 업데이트
- 역할 분담 및 일정 조율
- 개발 환경 구축(맵 디자인, SW 개발, 문서 통합 관리)
- 멀티 로봇 환경 구축 및 네비게이션
- 멀티 로봇 개별 업무 수행
- 멀티 로봇 협동 업무 수행
- (Optional) Turtlebot4 각종 센서 데이터의 이해와 적용

# DAY 8 (FINAL PROJECT)

---

- 시스템 설계에 기반한 객체 감지 모델 구현
  - 로봇 환경에 적용 및 Unit Test
  - 모듈로 제작하고 launch파일로 구현
  - code 정리 및 버전관리, 문서 작성 및 영상 촬영, 팀 내 기술 브리핑
- 시스템 설계에 기반한 SysMon 설계 구현
  - 로봇 환경에 적용 및 Unit Test
  - 모듈로 제작하고 launch파일로 구현
  - code 정리 및 버전관리, 문서 작성 및 영상 촬영, 팀 내 기술 브리핑

# DAY 8 (FINAL PROJECT)

---

- 시스템 설계에 기반한 AMR 제어 구현
- 로봇 환경에 적용 및 Unit Test
- 모듈로 제작하고 launch 파일로 구현
- code 정리 및 버전관리, 문서 작성 및 영상 촬영, 팀 내 기술 브리핑



# DAY 9 (FINAL PROJECT)

---

- 개별 기능 통합 구현 및 Integration 테스트
- 통합 Launch 파일로 구현
- Robust한 시스템 구축을 위한 예외 처리 및 Code Refactoring
- code 정리 및 버전관리, 문서 작성 및 영상 촬영, 팀 내 기술 브리핑

# DAY 10 (FINAL PROJECT)

---

- 프로젝트 발표 및 시연
- 최종 산출문 정리(소스코드, 발표 PPT, 동작 영상)
- 팀 간 기술 컨퍼런스를 통한 기술 극복 경험담, 노하우 교류(채점 대상X)

# PROJECT 평가

Mini Project	
1. 비즈니스 요구 사항 작성	5
2. 시스템 요구 사항 작성	5
3. Process Flow Diagram을 사용하여 시스템 설계를 생성	5
4. Detection Alert Module 상세설계 수행	5
5. Detection Alert Module의 코딩 및 테스트 수행	5
6. AMR Controller Module 상세설계 수행	5
7. AMR Controller Module의 코딩 및 테스트 수행	5
8. Detection Alert 및 AMR Controller Module의 통합 및 테스트 수행	5
9. System Monitor Module 상세설계 수행	5
10. System Monitor Module의 코딩 및 테스트 수행	5
11. 모든 모듈의 시스템 통합 및 테스트 수행	10
Mini Project Sub Total	60

12. 최종 프로젝트	
completeness	10
accuracy (object Det. / Goal arrival)	10
safety (obj. avoidance)	10
13. 최종 프로젝트 발표	10
14. System Design Doc	10
Final Project Sub Total	50

프로젝트 RULE NUMBER ONE!!!

---

Have Fun Fun Fun!





**BEGINS SPRINTS**  
**DETAIL DESIGN/CODING/TESTING**

---



# PROJECT SPRINTS

- Detection Alert

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

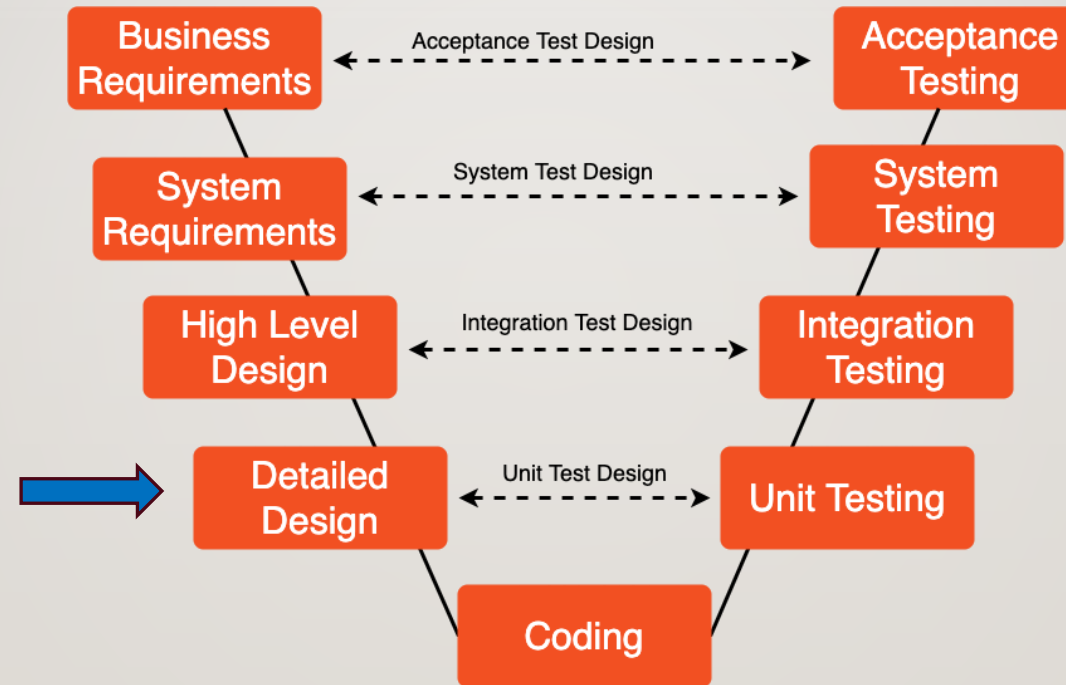
- AMR Controller

- 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

# SPRINT I - DETECTION ALERT



SDLC - V Model - notepub.io

# TEAM EXERCISE 4

---

Perform Detail Design of Detection Alert Module using Process Flow Diagram



# DETAIL DESIGN REVIEW BY EACH TEAM

---

Using the process flow diagram present team's design

# EXAMPLE DETAILED DESIGN DOCUMENT

## Detailed Design Document: AMR Navigation and Threat Detection

Project Title: Autonomous Mobile Robot (AMR) Security System

Version: 1.0

Date: [Insert Date]

### 1. Overview

This document outlines the detailed design for the Autonomous Mobile Robot (AMR) navigation and threat detection components. It covers the architecture, algorithms, data processing, and system interactions necessary to enable autonomous navigation within a secure area and real-time threat detection using onboard sensors.

### 2. System Architecture

The AMR system relies on onboard hardware (e.g., sensors, cameras, Jetson-Orin processor) and software (ROS2, OpenCV, YOLO) for autonomous navigation and real-time threat detection. All processing occurs locally on the AMR, with the capability to transmit alerts to a monitoring PC via Wi-Fi.

## 상세 설계 문서: AMR 네비게이션 및 위협 탐지

프로젝트 제목: 자율 이동 로봇(AMR) 보안 시스템

버전: 1.0

날짜: [날짜 삽입]

### 1. 개요

이 문서는 자율 이동 로봇(AMR)의 네비게이션 및 위협 탐지 구성 요소에 대한 상세 설계를 다룹니다. 자율 네비게이션과 실시간 위협 탐지를 위해 온보드 센서를 사용하는 데 필요한 아키텍처, 알고리즘, 데이터 처리 및 시스템 상호작용이 포함되어 있습니다.

### 2. 시스템 아키텍처

AMR 시스템은 자율 네비게이션 및 실시간 위협 탐지를 위해 온보드 하드웨어(예: 센서, 카메라, Jetson-Orin 프로세서)와 소프트웨어(ROS2, OpenCV, YOLO)를 활용합니다. 모든 처리는 AMR 내에서 로컬로 수행되며, 잠재적인 위협이 감지되면 Wi-Fi를 통해 모니터링 PC로 알림을 전송할 수 있습니다.

# 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

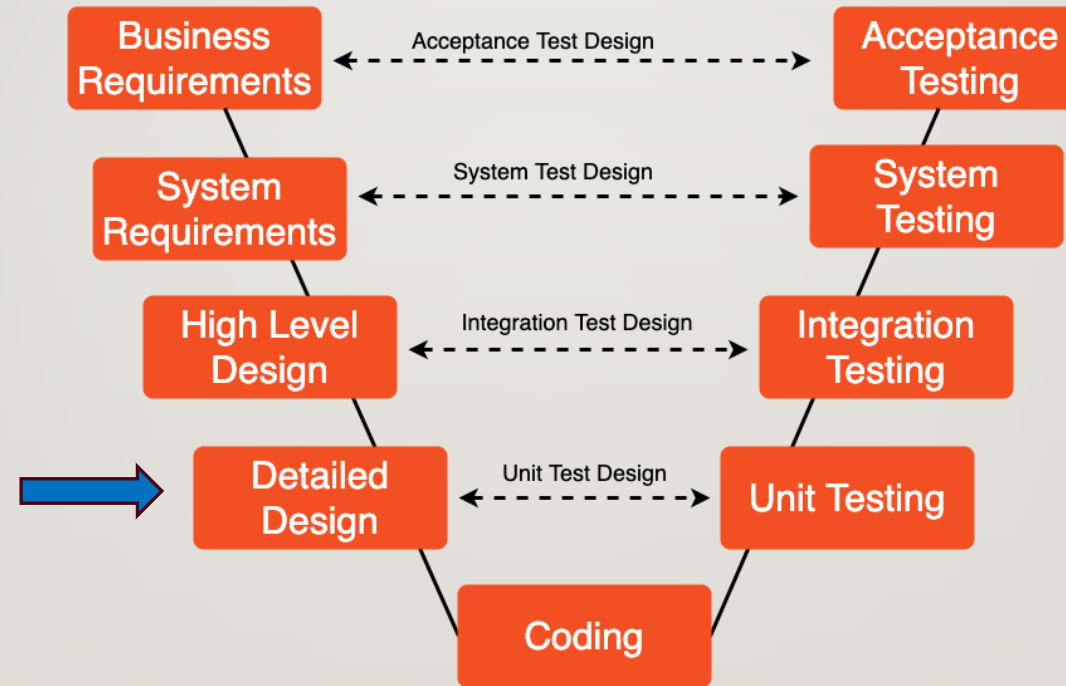
# AMR CONTROLLER SPRINT

---





# 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

# TODAY

---





# PROJECT SPRINTS

- Detection Alert

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

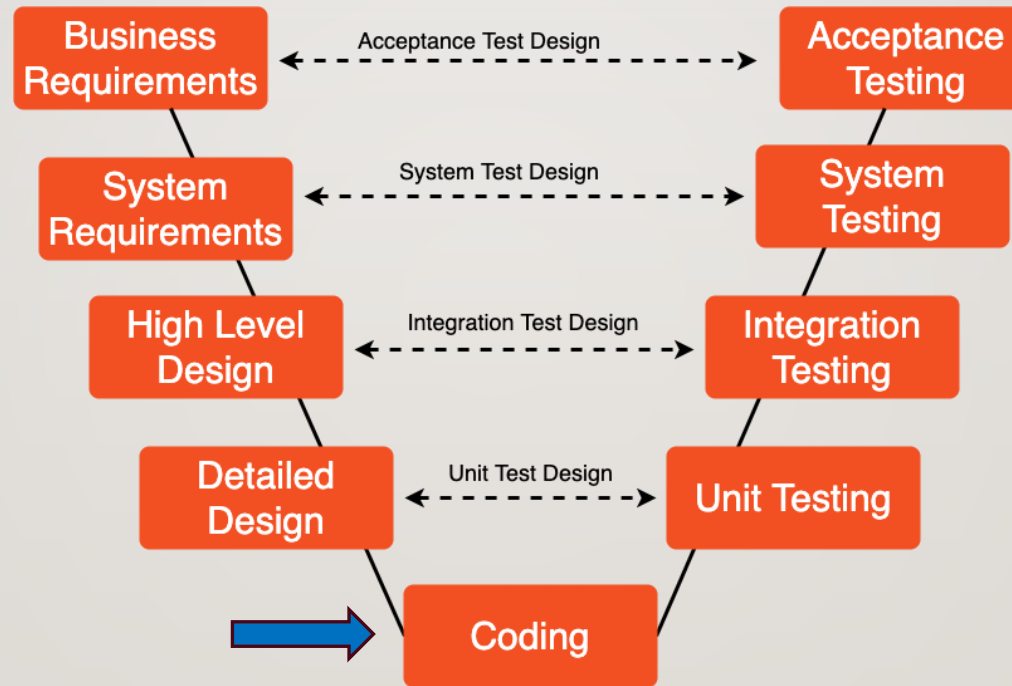
- AMR Controller

- 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

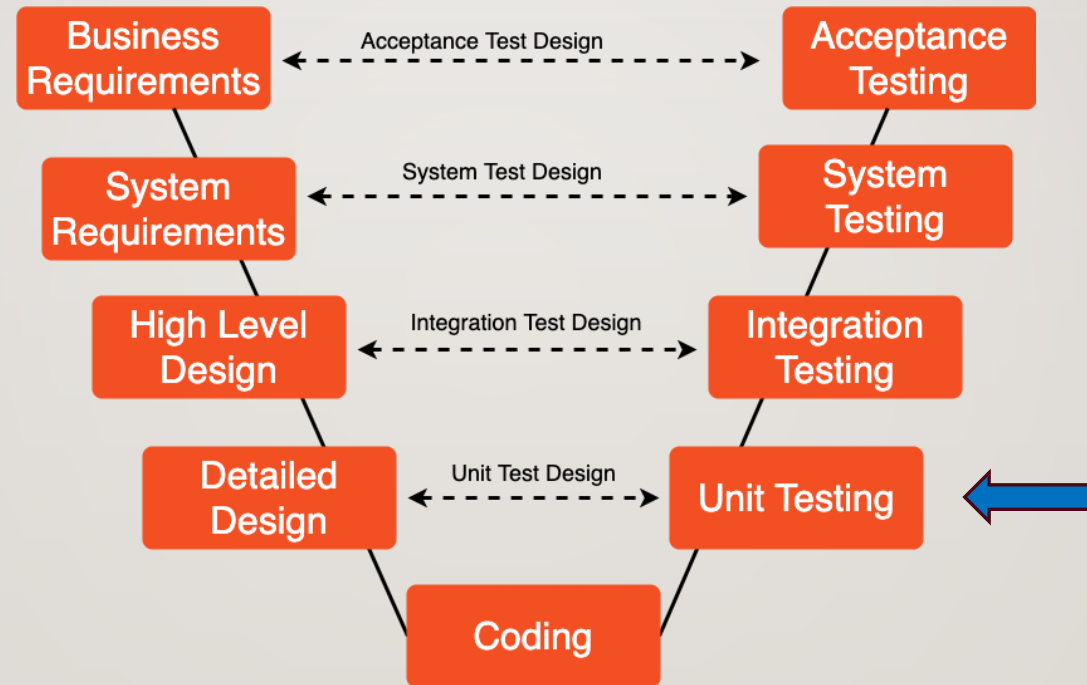
- Receive and Display Detection Camera and info
- Receive and Display AMR Camera and info
- Store, display, and report Information and Alerts

# SPRINT I - DETECTION ALERT



SDLC - V Model - notepub.io

# SPRINT I - DETECTION ALERT



SDLC - V Model - notepub.io

# TEAM EXERCISE 5

---

Perform coding and testing of Detection Alert Module



# EXPECTED OUTCOME

---

- Successful object detection
- ROS Nodes, and Topics created to send and display images and data

# 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

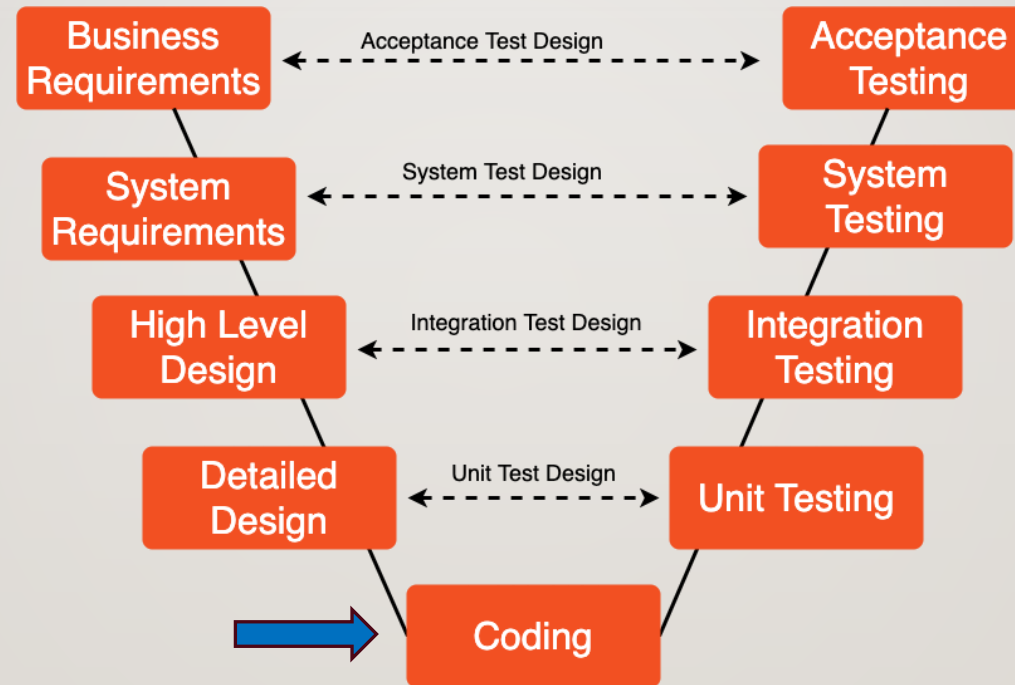
- AMR Controller

- 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

# SPRINT 2 – AMR CONTROLLER



SDLC - V Model - notepub.io

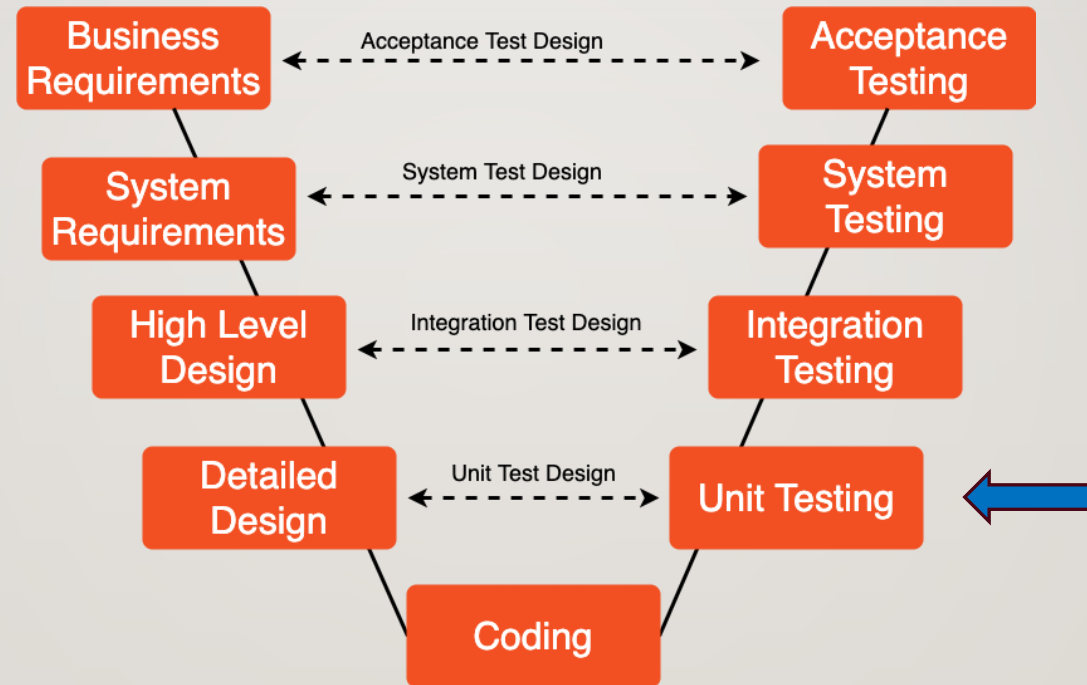


# EXPECTED OUTCOME

---

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

# SPRINT 2 – AMR CONTROLLER



SDLC - V Model - notepub.io

# TEAM EXERCISE 7

---

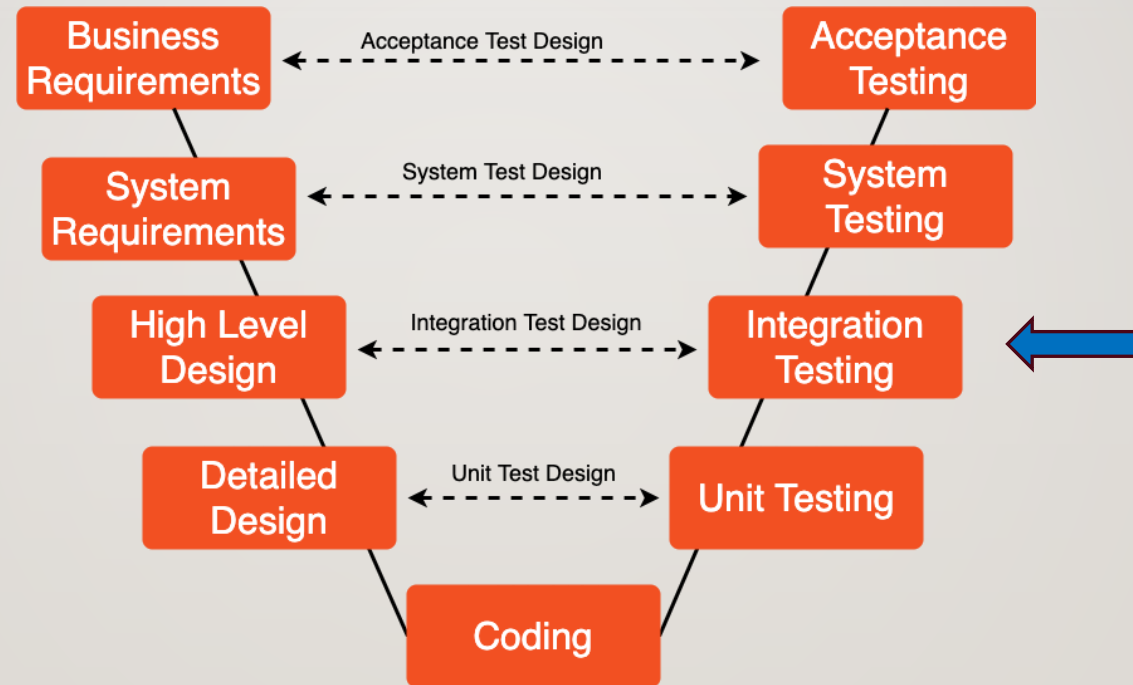
Perform coding and testing of AMR Controller Module

# 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

---

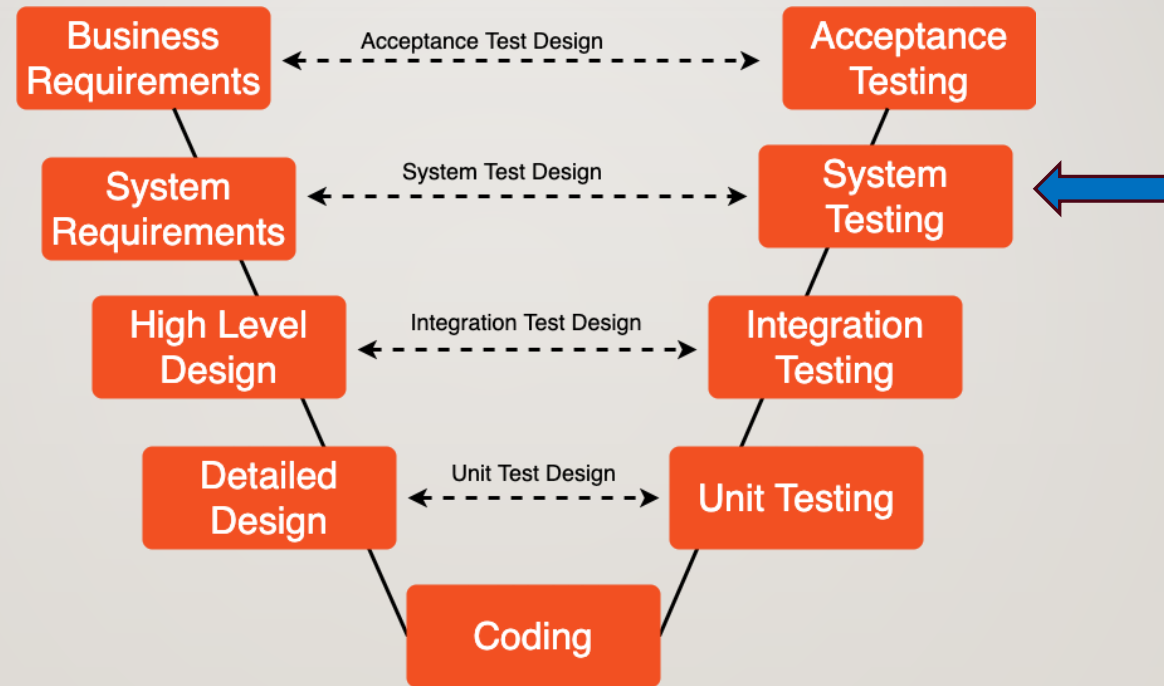
Perform integrate and test of Detection Alert and AMR Controller Modules

# RESULTS & CODE REVIEW BY EACH TEAM

---

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

# SYSTEM INTEGRATION & TEST



SDLC - V Model - notepub.io

# TEAM EXERCISE II

---

Perform integration and testing of Detection, AMR Controller, and System Monitor Module



# DEMONSTRATION OF SOLUTION BY EACH TEAM

---

Show actual results against the expected results and explain

# FINAL PROJECT

---



# 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

# 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

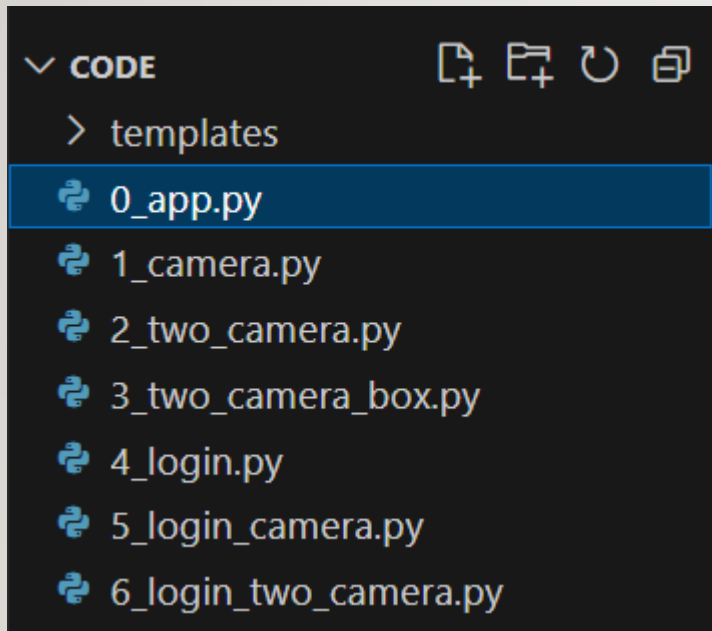
---

- 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

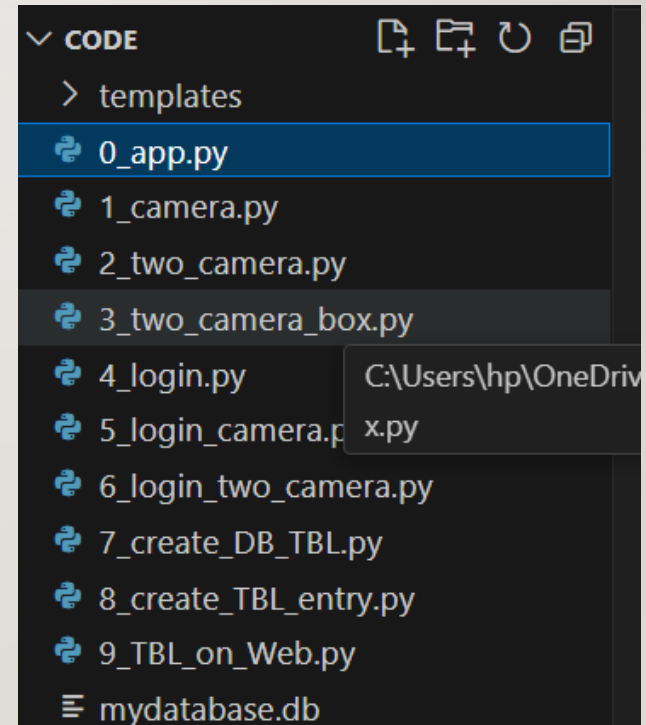
[CSS: Cascading Style Sheets | MDN](https://developer.mozilla.org/en-US/docs/Web/CSS)

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

# CODING HINTS

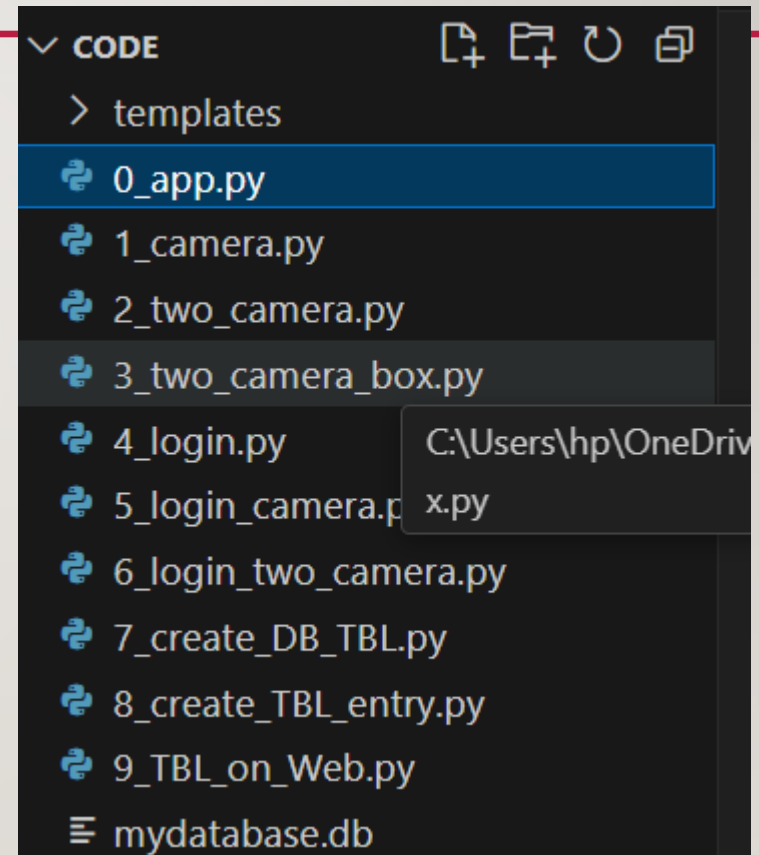
---

- Flask Basic Review
  - `sudo apt install sqlite3`
- SQLite Basic Review
  - SQLite is a lightweight, self-contained, serverless SQL database engine.



# CODING HINTS

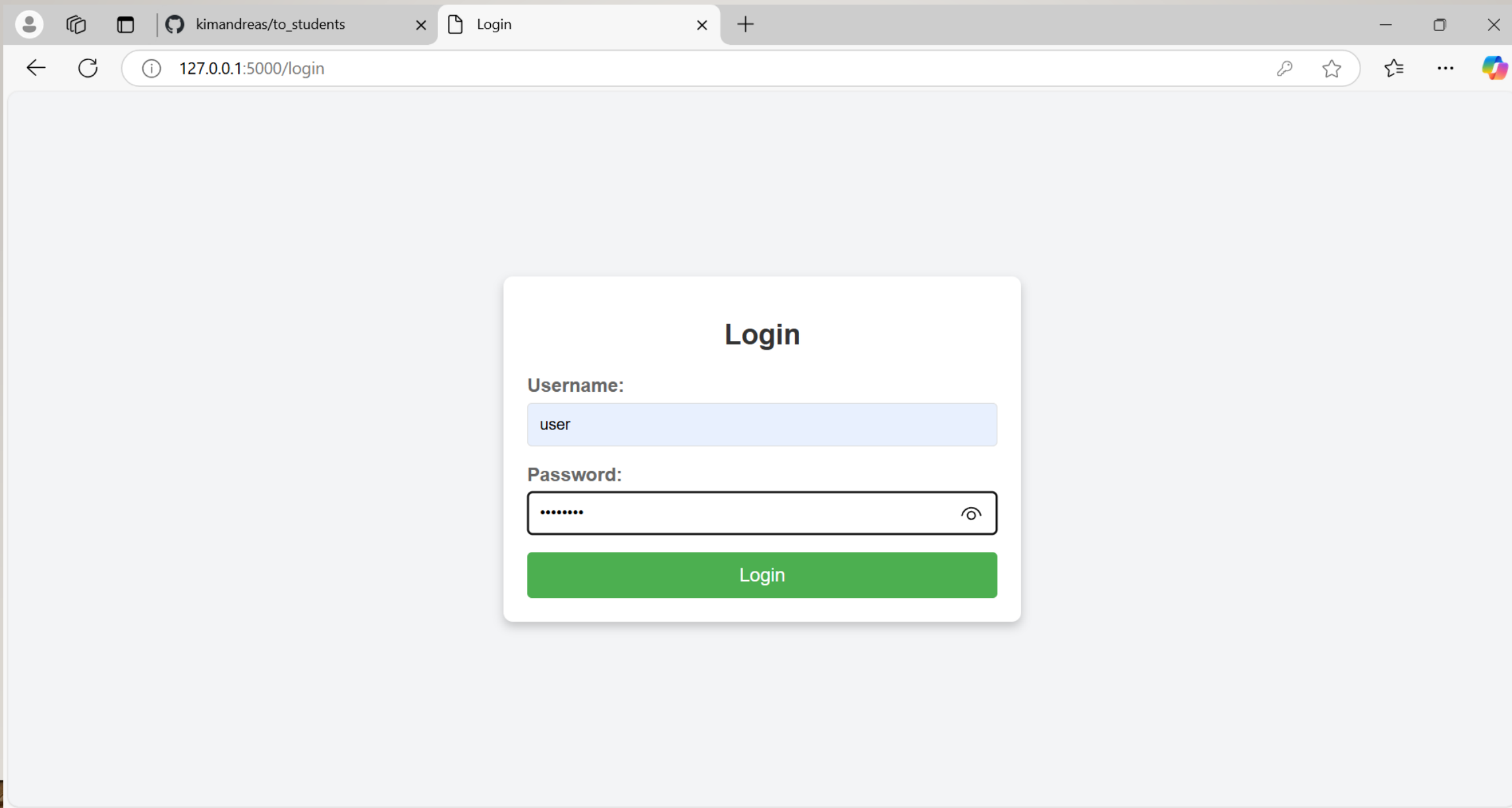
- Flask Basic Review
- SQLite Basic Review
- Webpage
  - Login page
  - Two video window
  - Alert Report
    - Status Captured and Following
- Database – SQLite
  - Detection Alert Data



# CODING HINTS

---

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





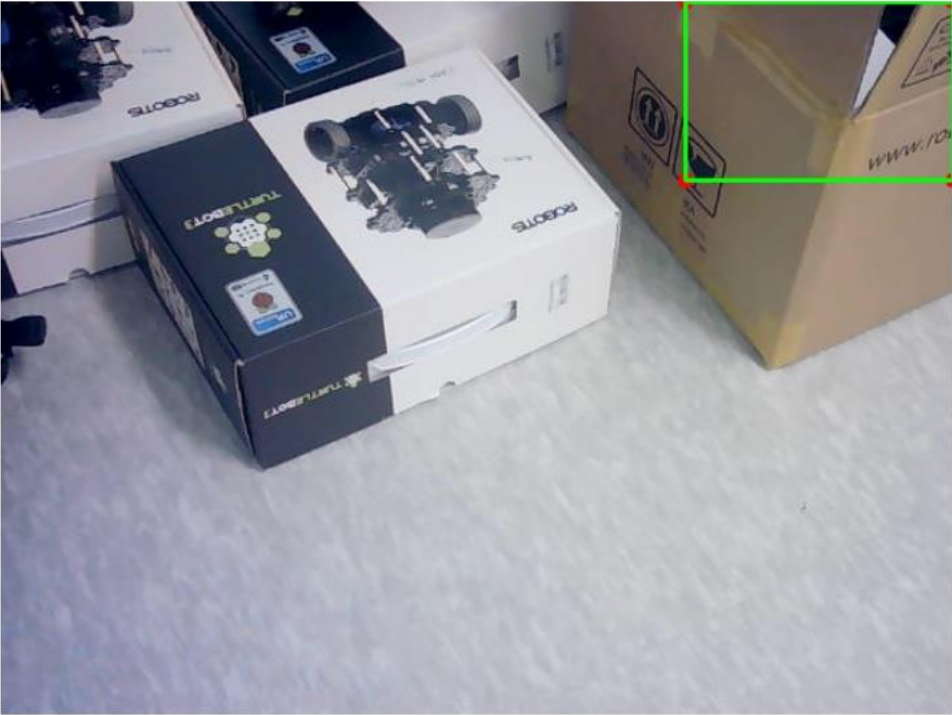
kimandreas/to\_students

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

# FINAL PROJECT TOPIC AND TEAM SELECTION

---



**BRAINSTORM A SITUATION THAT  
REQUIRES THIS SOLUTION**

---



# PROJECT JUSTIFICATION (WHY)

---

- Situation Analysis
  - evaluates both external and internal factors to determine the necessity and feasibility of a project. It helps justify resource allocation by outlining how the project aligns with strategic goals, identifying potential challenges and opportunities, and providing a detailed understanding of the project's context for informed decision-making.
- 상황 분석
  - 프로젝트의 필요성과 타당성을 결정하기 위해 외부 및 내부 요인을 모두 평가합니다. 프로젝트가 전략적 목표에 어떻게 부합하는지 설명하고, 잠재적인 과제와 기회를 식별하고, 정보에 입각한 의사 결정을 위해 프로젝트의 컨텍스트에 대한 자세한 이해를 제공하여 리소스 할당을 정당화하는 데 도움이 됩니다.



# PROJECT JUSTIFICATION (WHY)

---

- Business Needs/Pain Point Analysis

- identifies and assesses the problems and unmet needs of customers. This process helps businesses tailor their solutions to enhance customer satisfaction and loyalty by directly addressing these issues.

- 비즈니스 니즈/문제점 분석

- 문제와 충족되지 않은 요구를 식별하고 평가합니다. 이 프로세스는 기업이 이러한 문제를 직접 해결하여 고객 만족도와 충성도를 높일 수 있도록 솔루션을 맞춤화하는 데 도움이 됩니다.



# BRAINSTORMING RULES

---

- Every input is good input
- Do not critique inputs only seek to understand
- Organize inputs into logical groupings
- Sequence or show relationships as needed
- Use Posted Notes on Flip Chart



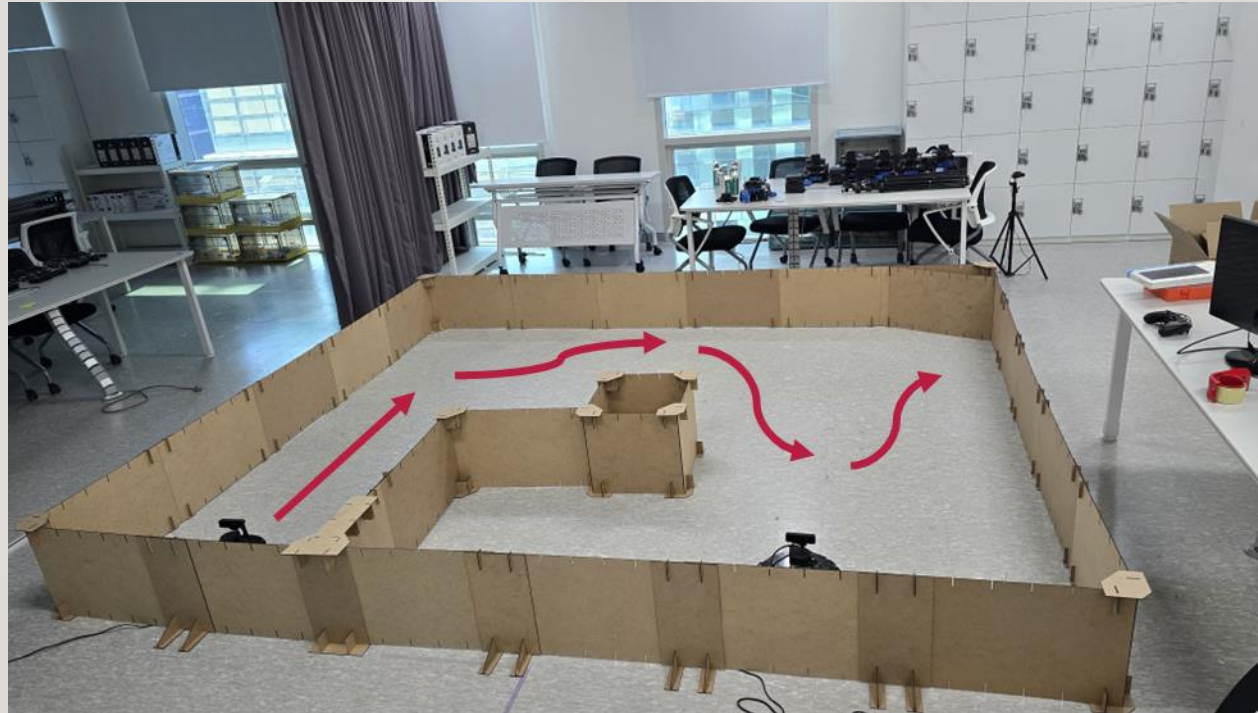
# DEVELOP YOUR BUSINESS SCENARIO (USE-CASE) PROCESS DIAGRAM

---

Using the posted notes and flipchart as needed

# SKETCH YOUR SCENARIO ON THE ENVIRONMENT

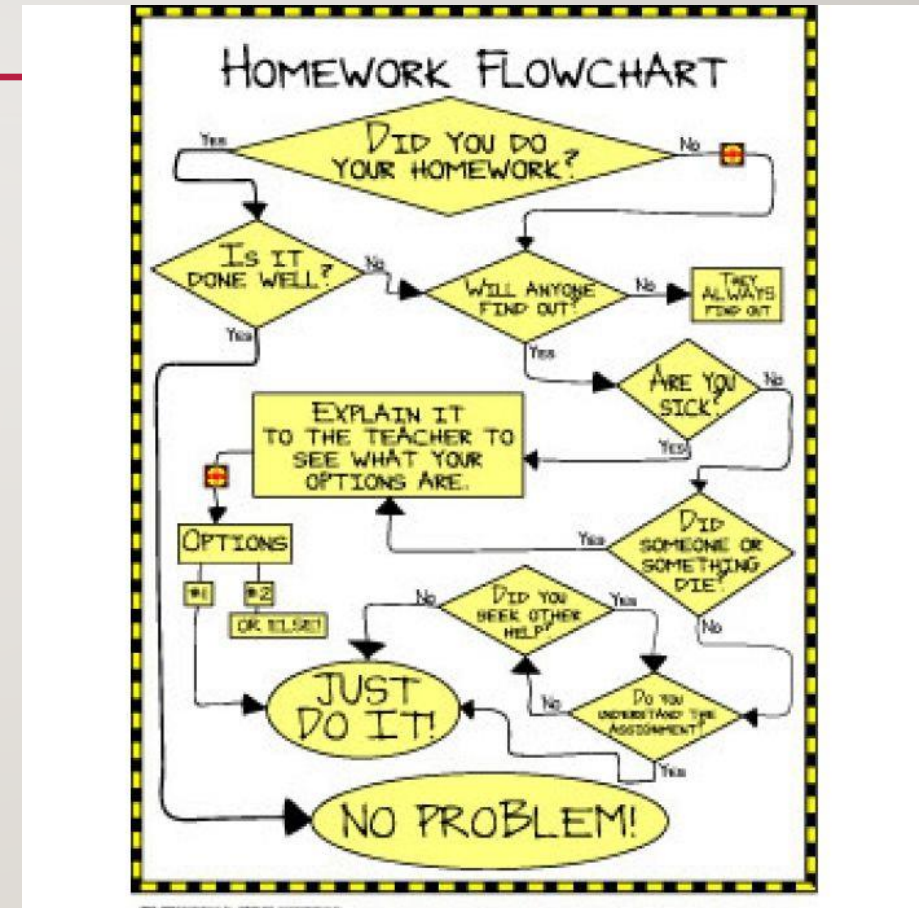
---





# VISUALIZATION – SCENARIO PROCESS DIAGRAMS

- As-Is Functional Process Diagram
  - Current states
- To-Be Functional Process Diagram
  - Future states
- [Untitled Diagram - draw.io](#)
- <https://app.diagrams.net/>



# BUSINESS REQUIREMENT (WHAT EXAMPLE)

---

- **Business Requirements with Metrics:** The company aims to deploy a robotic system integrated with a deep learning model to automate quality inspection in manufacturing. The goal is to reduce human error by achieving 98% accuracy in defect detection and increase production efficiency by minimizing inspection time to under 2 seconds per item.
- 이 회사는 딥 러닝 모델과 통합된 로봇 시스템을 배포하여 제조 시 품질 검사를 자동화하는 것을 목표로 합니다. 목표는 결함 감지에서 98%의 정확도를 달성하여 인적 오류를 줄이고 검사 시간을 품목당 2초 미만으로 최소화하여 생산 효율성을 높이는 것입니다.



# TEAM EXERCISE I

---

Brainstorm Business Requirement for the project and write business requirement statement

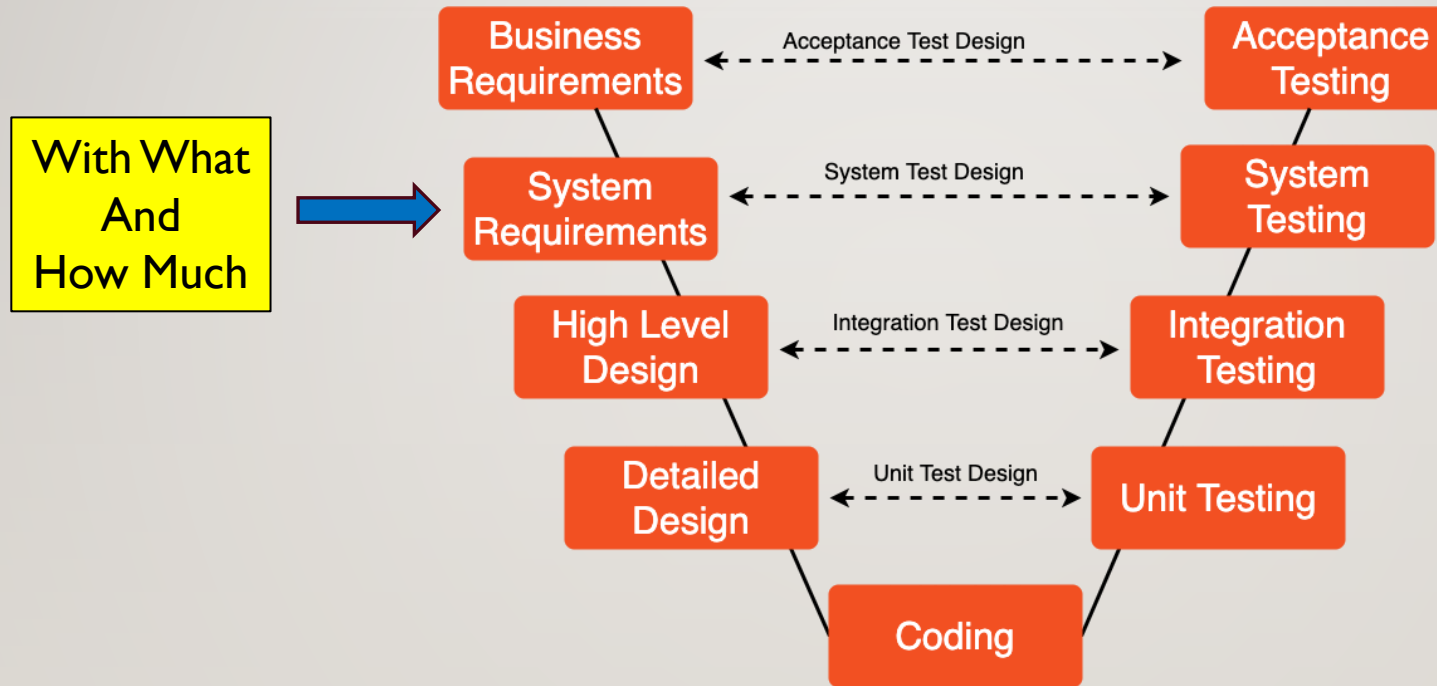
Using the posted notes and flipchart as needed

# BUSINESS REQUIREMENT PRESENTATION BY EACH TEAM

---

Using the posted notes and flipchart as needed

# SW DEVELOPMENT PROCESS



SDLC - V Model - notepub.io

# BASE HW/OS X 2!!

---

- PC

- Ubuntu 22.04
- USB Camera



- Network
  - Wifi



- AMR

- TurtleBot4
- Ubuntu 22.04





# MULTI-ROBOT SETUP - PC

---

- [Multi Robot Setup](#)
- <https://indecisive-freedom-6e8.notion.site/Multi-Robot-Setup-1e98e215779c807d9918cd2a0bd8fa01>



# OBJ. DET. X 2

---

TARGET



DUMMY



# TEAM EXERCISE 2

---

Brainstorm **Updated** System Requirement for the project and document

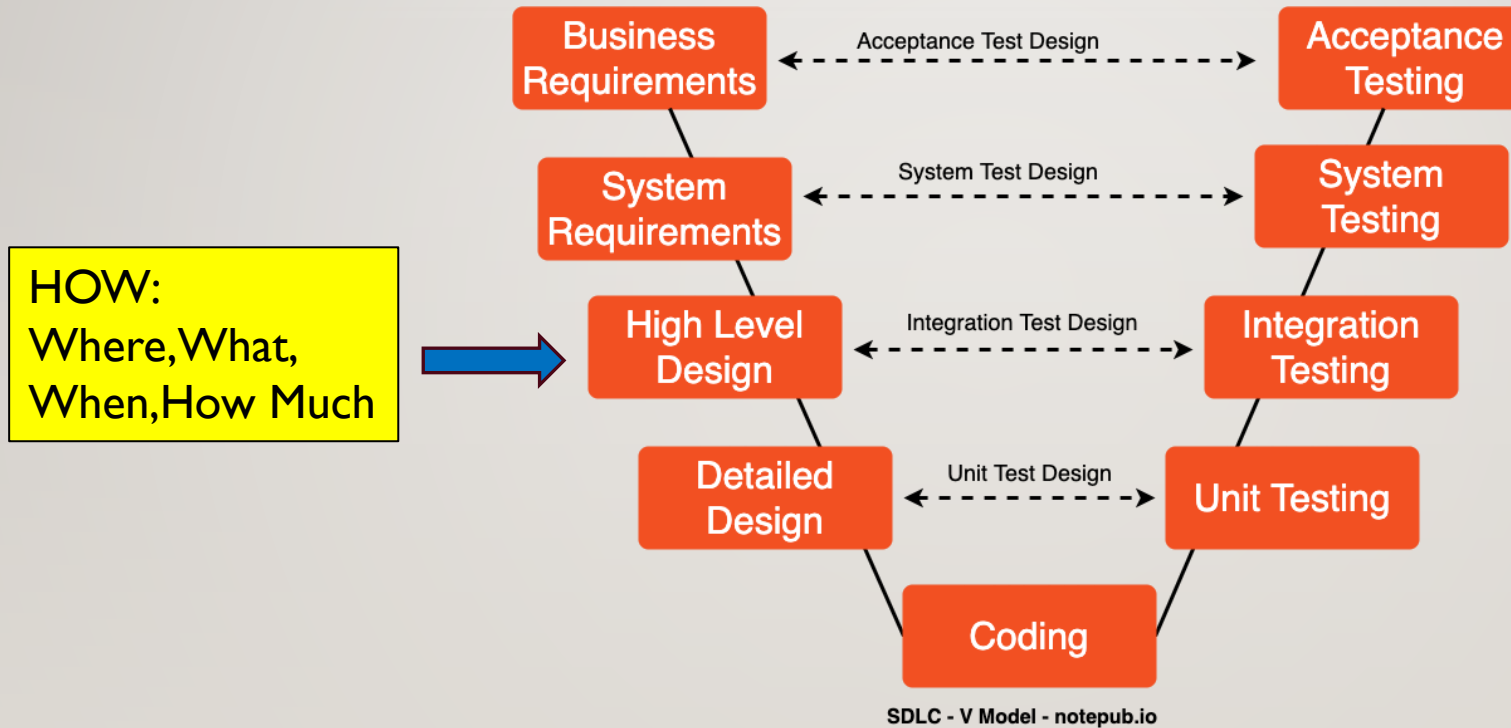
Using the posted notes and flipchart as needed

# SYSTEM REQUIREMENT PRESENTATION BY EACH TEAM

---

Using the posted notes and flipchart as needed

# SW DEVELOPMENT PROCESS





# KEY SUBSYSTEM (MODULES) TO DEVELOP

---

- Detection ???
- AMR Controller ???
- System Monitor ???

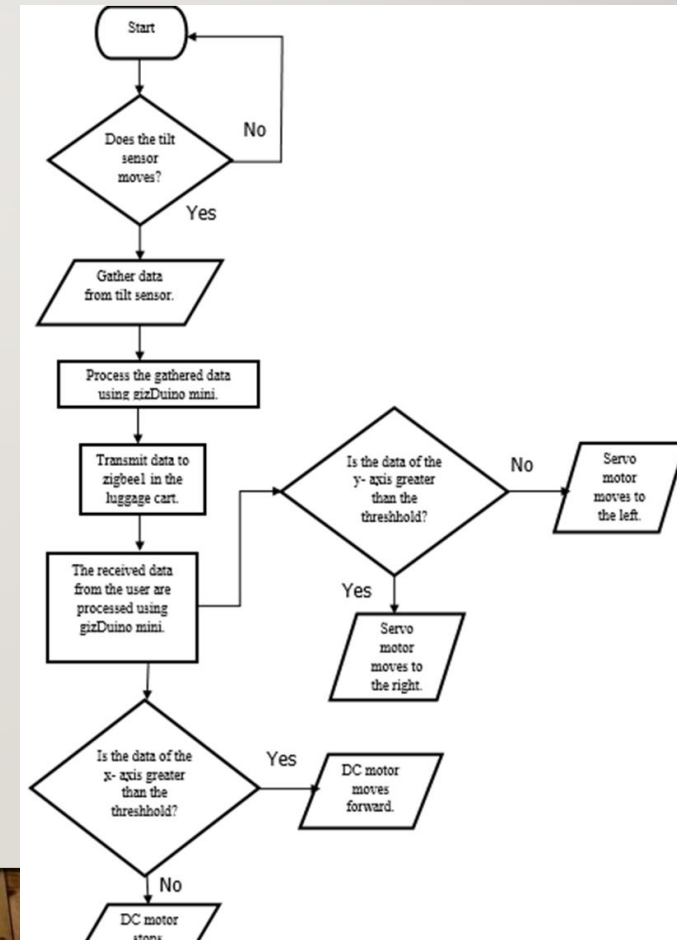


# VISUALIZATION – SYSTEM FUNCTIONAL PROCESS FLOW DIAGRAMS

- To-Be Functional Process Flow Diagram

Detection Alert  
System Monitor  
AMR Controller

- **F**unctions
- **I**nterfaces  
Dataflow
- **T**esting  
Error and Exception Handling



# TEAM EXERCISE 3

---

Create System Design using Process Flow Diagram.

Use the posted notes and flipchart as needed

# SYSTEM DESIGN PRESENTATION BY EACH TEAM

---



# EXAMPLE SYSTEM DESIGN DOCUMENT

## System Design Document (SDD)❧

Project Title: Autonomous Mobile Robot (AMR) Security System↓

Version: 1.1↓

Date: [Insert Date]❧

### 1. Overview❧

The Autonomous Mobile Robot (AMR) Security System is designed to provide autonomous patrolling, threat detection, and alerting within a secure area using a single AI-enabled robot. The system consists of one AMR equipped with necessary hardware and software components to operate independently, processing data on-board without the need for a central server.❧

### 2. System Architecture❧

Since the system consists of a single AMR, data processing, navigation, threat detection, and alerting are all performed locally on the AMR itself. The AMR communicates directly with a user interface on a PC via a local network (Wi-Fi) for monitoring, alerts, and manual override if required.❧

## 시스템 설계 문서 (SDD)❧

프로젝트 제목: 자율 이동 로봇(AMR) 보안 시스템↓

버전: 1.1↓

날짜: [날짜 삽입]❧

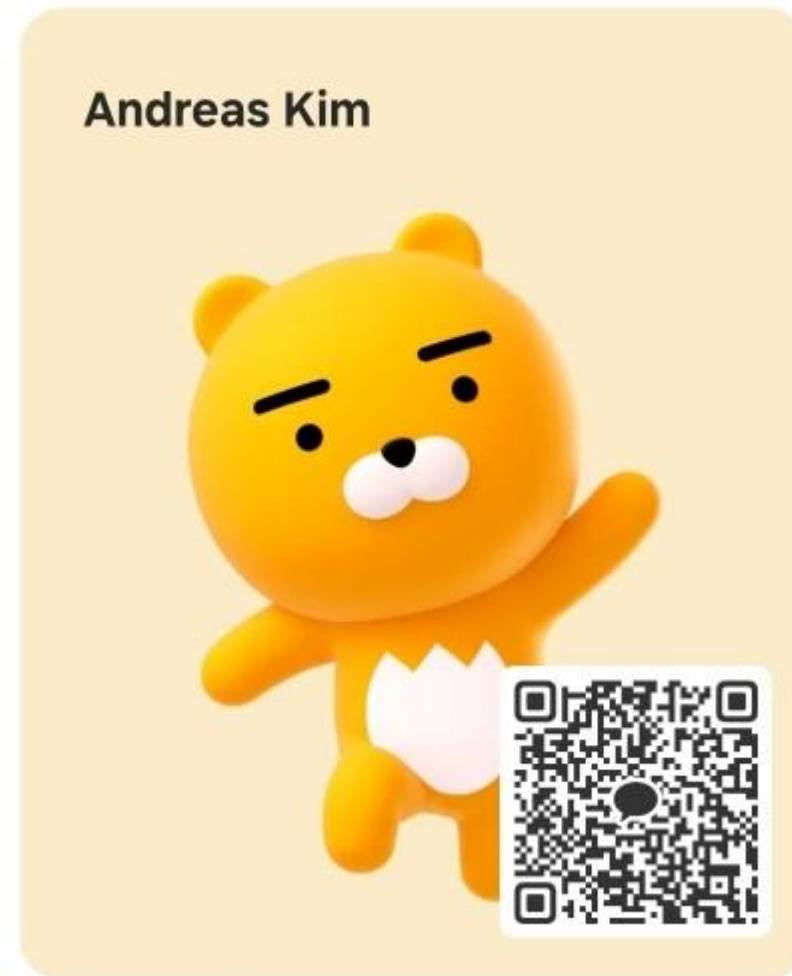
### 1. 개요❧

자율 이동 로봇(AMR) 보안 시스템은 단일 AI 기반 로봇을 사용하여 보안 구역 내에서 자율 순찰, 위협 탐지 및 경고를 제공하도록 설계되었습니다. 시스템은 단일 AMR이 독립적으로 작동할 수 있도록 필요한 하드웨어 및 소프트웨어 구성 요소로 구성되며, 중앙 서버 없이 데이터를 현장에서 처리합니다.❧

### 2. 시스템 아키텍처❧

이 시스템은 단일 AMR으로 구성되므로 데이터 처리, 네비게이션, 위협 탐지 및 경고가 모두 AMR에서 로컬로 수행됩니다. AMR은 모니터링, 알림 및 수동 제어를 위해 PC의 사용자 인터페이스와 로컬 네트워크(Wi-Fi)를 통해 직접 통신합니다.❧

Send System Design Doc.  
Here:





# PROJECT TIMELINE/CRITICAL PATH ITEM MANAGEMENT

---



# EX. IMPLEMENTATION TIMELINE

Function Backlog	Owner	5월 20일	5월 21일	5월 22일	5월 23일	5월 24일	5월 25일
<b>Unloading Module</b>	John						
Input1	John						
Input2	John						
Output 1	John						
Unit Test	John						
<b>Receiving Module</b>	Jan						
Input1	Feb						
Input2	Mar						
Output 1	Apr						
Unit Test	John						
Integration Test	John/Jan						

이 타임라인을 생성할 때  
먼저 시스템 및 시스템  
설계의 기능 프로세스  
다이어그램(To-Be)을  
완료해야 합니다.

그런 다음 각 기능(하위  
함수/모듈 및  
입력/출력)에 대해 누가,  
무엇을, 언제, 어떻게  
정의합니다. 표에 설명  
타임라인 형식의 무엇을,  
누가, 언제를 입력합니다.

# 프로젝트 RULE NUMBER ONE!!!

---

Are we having  
Fun???

