F-300

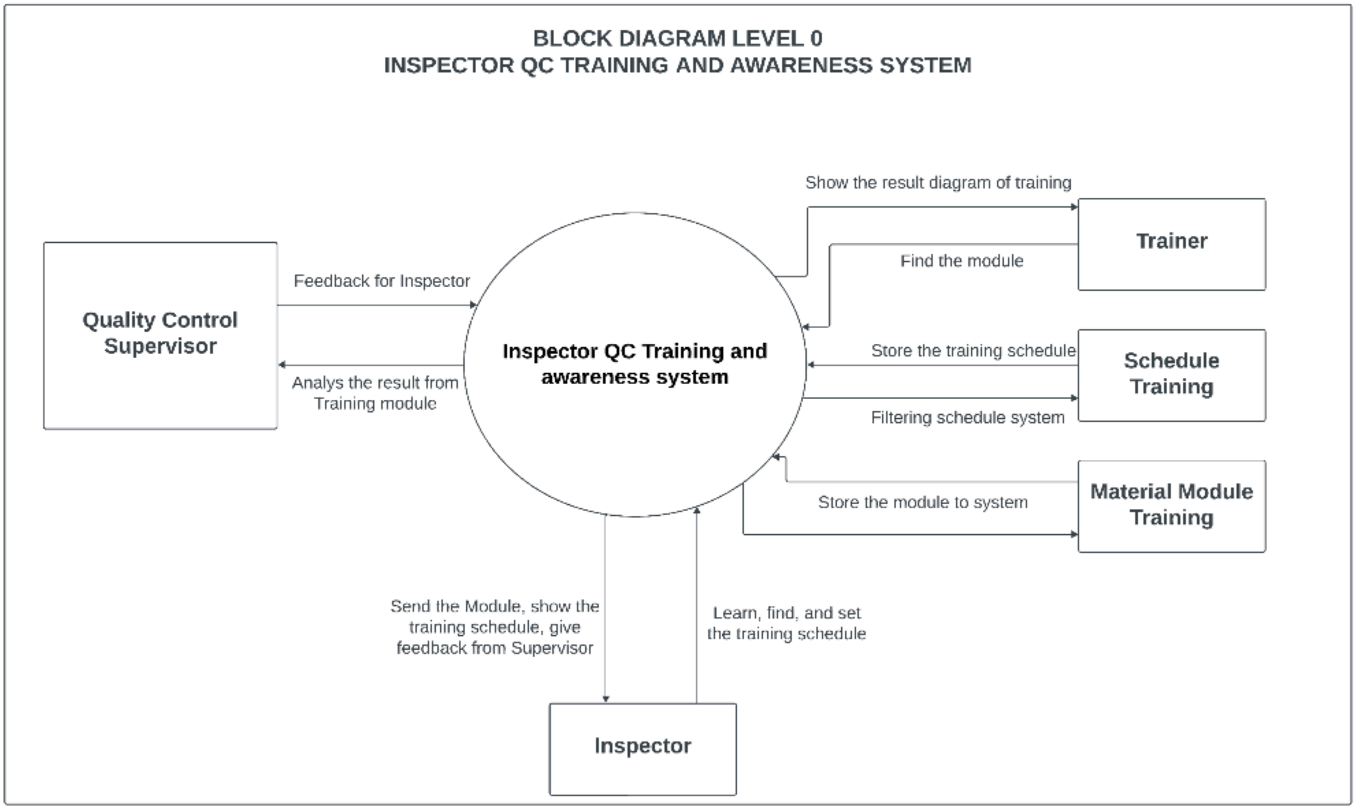
DESIGN

3.1 Alternative Solution Designs

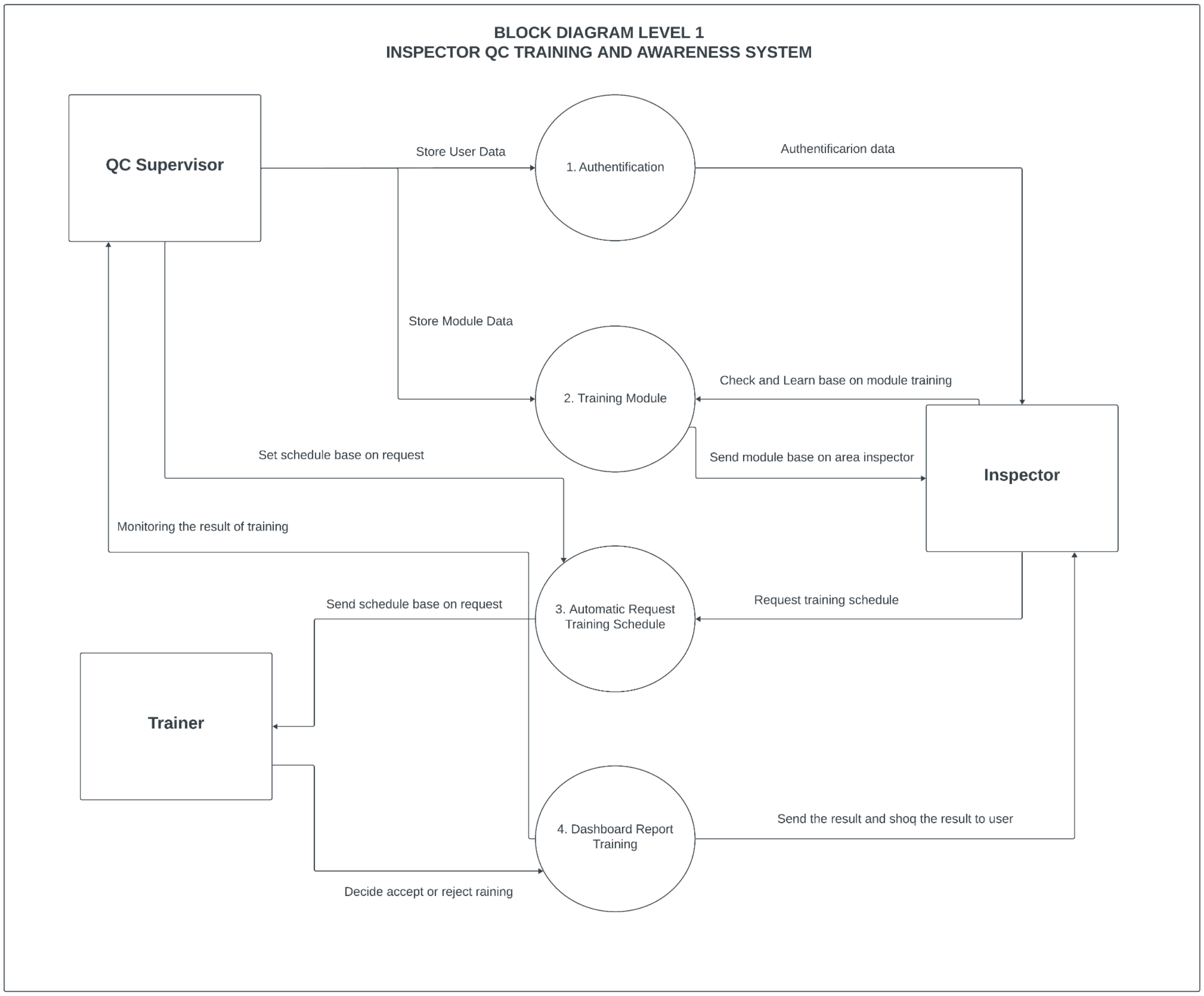
3.1.1 Alternative Solution 1: Inspector Training and awareness

Inspector training and awareness is awareness-raising programs designed to improve inspectors in the field understanding the manufacturing environment, starting from work safety until the quality check technique and procedures. This program aims to provide knowledge and skills to inspectors in the field so that they can improve their performance during work.

Block diagram level 0 :


BLOCK DIAGRAM LEVEL 0
INSPECTOR QC TRAINING AND AWARENESS SYSTEM
Quality Control Supervisor
Feedback for Inspector
Analys the result from Training module
Inspector QC Training and awareness system
Show the result diagram of training
Find the module
Trainer
Store the training schedule
Filtering schedule system
Schedule Training
Store the module to system
Material Module Training
Send the Module, show the training schedule, give feedback from Supervisor
Learn, find, and set the training schedule
Inspector

Block diagram level 1 :


BLOCK DIAGRAM LEVEL 1
INSPECTOR QC TRAINING AND AWARENESS SYSTEM
QC Supervisor
Store User Data
1. Authentification
Authentificarion data
Store Module Data
2. Training Module
Check and Learn base on module training
Send module base on area inspector
Inspector
Set schedule base on request
Monitoring the result of training
3. Automatic Request Training Schedule
Send schedule base on request
Request training schedule
Trainer
4. Dashboard Report Training
Send the result and shoq the result to user
Decide accept or reject raining

Web Interface Component :

- Username User Profile

  The User Profile area of the interface holds all the individual's pertinent information which includes the full name, ID number, office location, role, and department within the company. In addition to that, the user profile photo is provided also. This part improves the user experience in customizing it, simplifying identification and encouraging productive communication and teamwork. To ensure that the system is current and engaging, users have the ability to modify their profiles, and this makes the system more human centered, agile and dynamic.

- Automation Dashboard

  This dashboard automation is used to allow admin show trainer reports that are input in a database and then it will be converted to an automated dashboard displaying users statistics.

- Feedback and evaluation

  Feedback and evaluation components will have a material survey where employees should fill each time when they lead a session. The survey will be rather like a small quiz on the session's material.

- Training Material

  Comprehensive instructional tools and training materials are under the Training Material Area of the web interface. A huge variety of educational resources in the form of documents, presentations, and multimedia content designed for developing knowledge and skills are accessible to users. Through an online base of learning assets, this feature facilitates the learning process by using training materials targeted at supporting users' professional development.

- Training Schedule

  A structured overview of upcoming training sessions and activities is offered by the Training Schedule functionality of the web interface. It enables users to see the dates, hours and locations of scheduled training sessions that increases the participation rate and planning. This feature also promotes organizational openness regarding skill development programmes as well as to inform users of learning options.

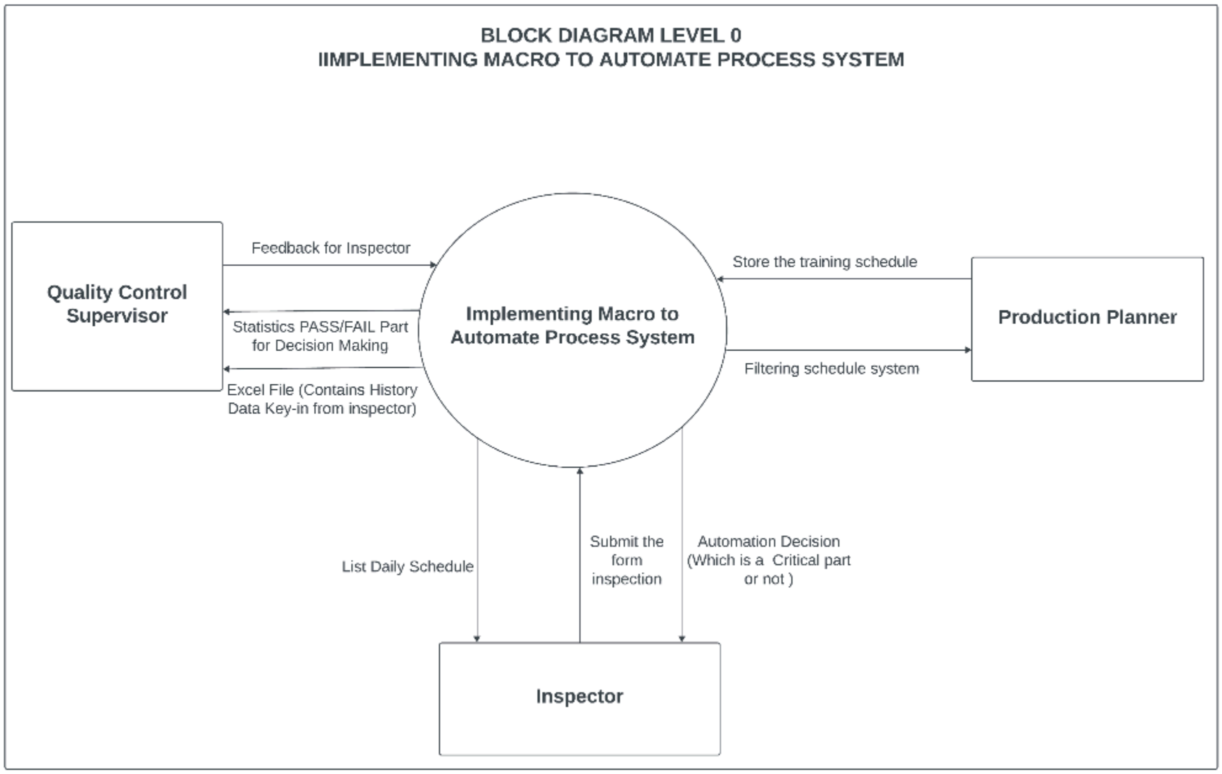
- **Algorithm for Alternative Solution 1: Inspector Training and Awareness**

The algorithm encompasses user profile management, facilitating personalized experiences by displaying essential information and allowing customization. The automation dashboard streamlines data entry and generates statistics, providing insights into training effectiveness. Feedback and evaluation involve material surveys to gauge understanding, enhancing the training's impact. The training material area offers organized resources, enabling quick access to documents and multimedia content for efficient learning. The training schedule feature provides a detailed overview of upcoming sessions, fostering participation and organizational transparency in skill development programs.

3.1.2 Alternative Solution 2 : Implementing Macro to Automate Process

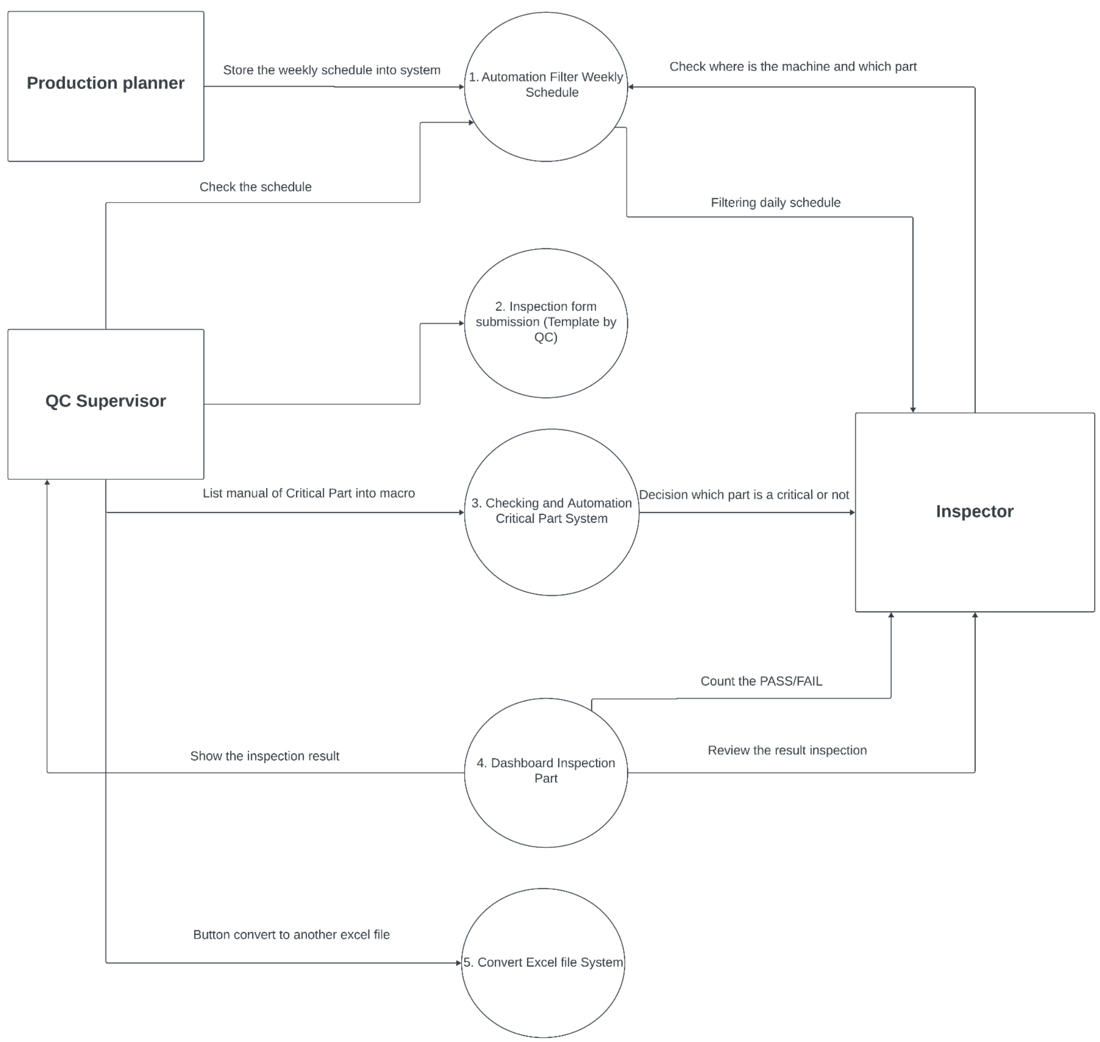
Using a Macro for the Quality Control business process will make the tasks easier and faster for inspectors on the field. It is like a shortcut or a tool that does a lot of the work for them. This means the inspector will not have to do as much manual work, saving them time and effort.

Block diagram level 0 :


BLOCK DIAGRAM LEVEL 0
IIMPLEMENTING MACRO TO AUTOMATE PROCESS SYSTEM
Quality Control Supervisor
Feedback for Inspector
Statistics PASS/FAIL Part for Decision Making
Excel File (Contains History Data Key-in from inspector)
Implementing Macro to Automate Process System
Store the training schedule
Filtering schedule system
Production Planner
List Daily Schedule
Submit the form inspection
Automation Decision (Which is a Critical part or not)
Inspector

Block diagram level 1 :

BLOCK DIAGRAM LEVEL 1
IMPLEMENTING MACRO TO AUTOMATE PROCESS SYSTEM


Production planner
Store the weekly schedule into system
1. Automation Filter Weekly Schedule
Check where is the machine and which part
Check the schedule
Filtering daily schedule
2. Inspection form submission (Template by QC)
QC Supervisor
Inspector
List manual of Critical Part into macro
3. Checking and Automation Critical Part System
Decision which part is a critical or not
Count the PASS/FAIL
Show the inspection result
4. Dashboard Inspection Part
Review the result inspection
Button convert to another excel file
5. Convert Excel file System

Excel Macro Interface Explanation:

The Quality Control Automation System with Macro Implementation introduces a sophisticated approach to streamline the inspection process, incorporating three integral databases. The Critical Part Database serves as a repository for data on parts with heightened inspection risks, ensuring inspectors are well-versed in specific material requirements. Simultaneously, the Checking Quality Performance Database scrutinizes user-generated inputs, systematically assessing parts for potential risks in subsequent inspections. Complementing these, the History Inspection Database meticulously archives the outcomes of inspections, enabling historical tracking and insightful analysis.

This comprehensive system converges its databases to generate an Automation Decision Result Output through macro implementation. As inspectors input data into the Critical Part Database and Checking Quality Performance Database, the macro intelligently analyzes this information. It factors in historical data from the History Inspection Database, facilitating a nuanced and informed decision-making process. The ultimate output categorizes inspected parts as either PASS or FAIL, providing inspectors with prompt and reliable decisions. This innovative system not only enhances the efficiency of the quality control process but also empowers inspectors with a robust decision support mechanism for their daily operations.

➢ **Algorithm for Alternative Solution 2: Implementing Macro to Automate Process**

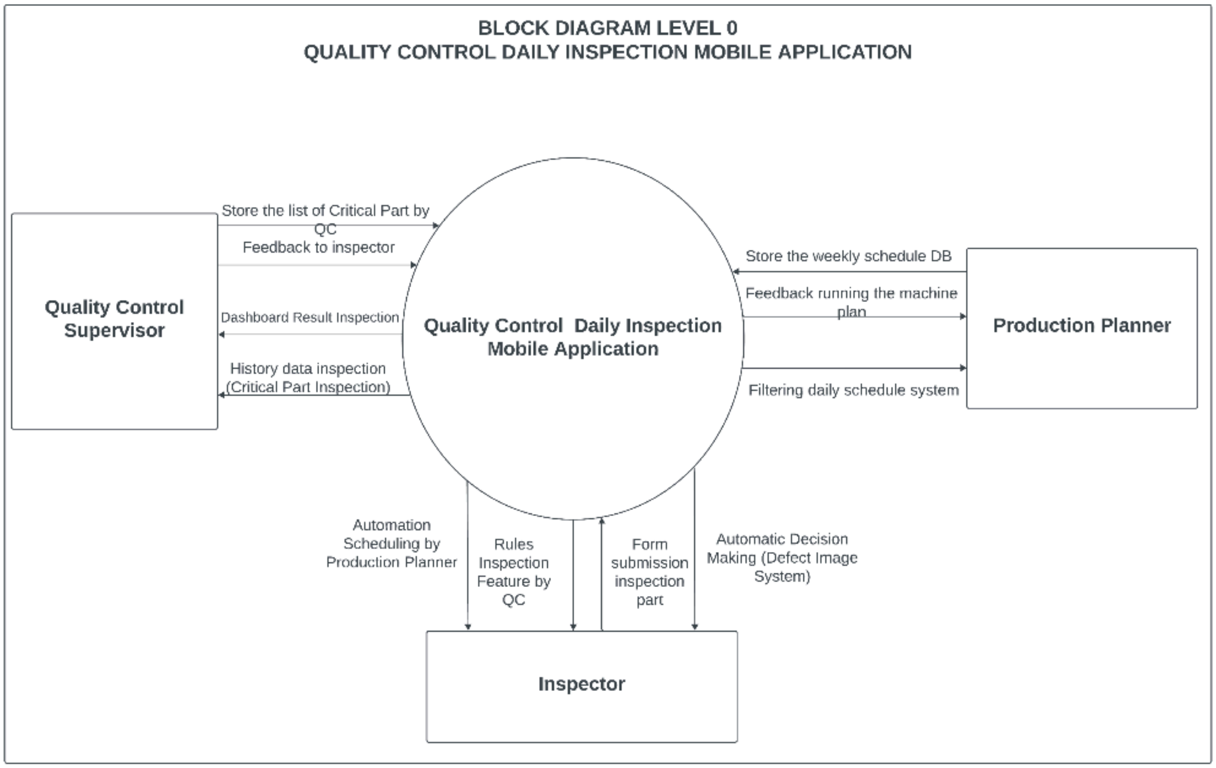
The inspectors record the data which they enter into the Critical Part Database that highlights edges with higher inspection risks increasing their understanding what material requirements are actually for specific parts. Simultaneously with this, aspired user-generated feedback is systematically analyzed and assessed in the Checking Quality Performance Database, therefore saving time and ultimately reducing the number of inspections and the likelihood of risks. The History Inspection Database preserves the final results of detailed procedures in the past, serving as a valuable source for decision-making. The macro rigorously evaluates information from both databases featuring trending data as a reference to decipher the inspected items as PASS or FAIL.

This is a system that offers efficient and reliable decisions to the inspectors making their everyday work easy and successful through a handy decision-support tool.

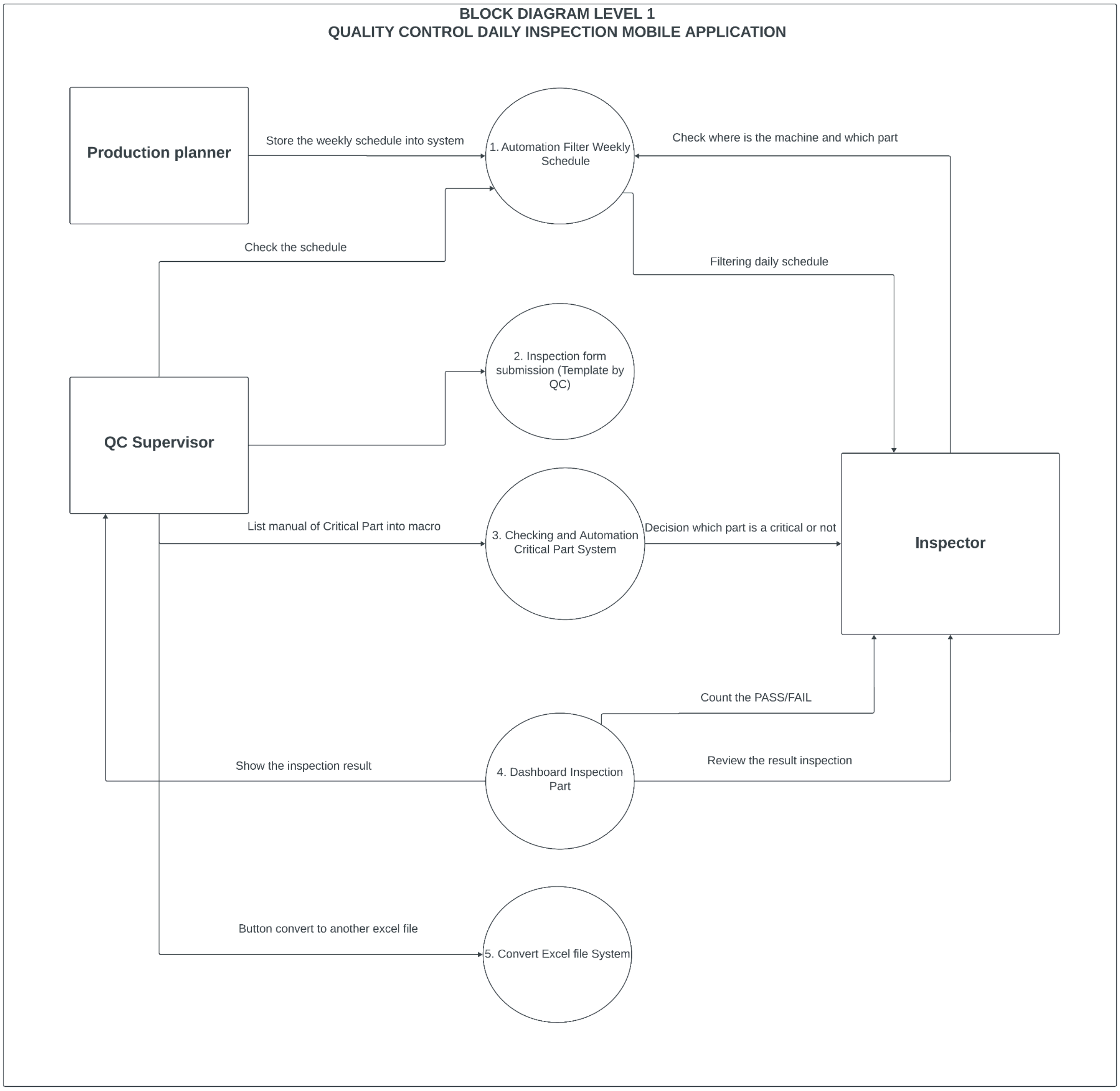
3.1.3 Alternative Solution 3: Quality Control Mobile Application

Quality Control Mobile App is such a device on a phone or tablet that aids in inspecting and sustaining products up to the standard. Using this app the quality check process is being accelerated, there are fewer mistakes being made by the inspectors and it is very convenient for the inspectors because they can carry the app around the manufacturing site.

Block diagram level 0 :


BLOCK DIAGRAM LEVEL 0
QUALITY CONTROL DAILY INSPECTION MOBILE APPLICATION
Quality Control Supervisor
Store the list of Critical Part by QC
Feedback to inspector
Dashboard Result Inspection
History data inspection (Critical Part Inspection)
Quality Control Daily Inspection Mobile Application
Store the weekly schedule DB
Feedback running the machine plan
Filtering daily schedule system
Production Planner
Automation Scheduling by Production Planner
Rules Inspection Feature by QC
Form submission inspection part
Automatic Decision Making (Defect Image System)
Inspector

Block diagram level 1 :


BLOCK DIAGRAM LEVEL 1
QUALITY CONTROL DAILY INSPECTION MOBILE APPLICATION
Production planner
Store the weekly schedule into system
1. Automation Filter Weekly Schedule
Check where is the machine and which part
Check the schedule
Filtering daily schedule
2. Inspection form submission (Template by QC)
QC Supervisor
List manual of Critical Part into macro
3. Checking and Automation Critical Part System
Decision which part is a critical or not
Inspector
Count the PASS/FAIL
Show the inspection result
4. Dashboard Inspection Part
Review the result inspection
Button convert to another excel file
5. Convert Excel file System

In this Mobile Application there are several features that are in it and are quite helpful in carrying out the daily inspection process which is carried out every day by the QC Team. In this application, there are several features that are very superior, such as:

Features 1: Scheduling and Decision System Based on Mobile Application

This module automates the production scheduling and inspection decision-making. The database integrates it for fast access to the production schedule where each part and the relevant machine location and past inspection records are found. It is worthy of note that it permits direct entry of the inspection reports which eliminates the possibility of human mistake. This also offers flexibility for the different industries so that during inspections faster and better decisions can be made.

Features 2: The Automatic Scanner Features is part of the Indicator.

By integrating part scanning into the automation, it takes the inspection efficiency to a new level. It offers crucial information such as the inspection time, material composition, and also part type. Real time inspection times help in the workflow planning while information about the materials and machine learning categorization are used to improve the inventory management. The user-friendly interface offers a smooth navigation, which is flexible to different workflow setup, hence improved efficiency of the operation.

Features 3: Automatic Product Defect Detection by Camera

This module can achieve automated defect detection using its camera. Fast scanning allows the detection of faulty areas, with high precision highlighting for the immediate fault identification. Through this automation the inspection process is lifted to a high level and the quality assessment of the product to a fast pace. Through automation of some manual activities, it provides thorough quality assurance, also eliminates mistakes and improves overall product quality control.

3.2 Rational/Systematic Design

3.2.1 Comparison Table

To determine the best solution among the three options, we need to compare them in detail across five criteria. Each criteria is important in figuring out which solution will be most effective in solving the problems, below is an explanation of each criteria:

a. Software Requirement: The necessary software tools for each solution, including interface design, backend development, and any additional dependencies.
b. Hardware Requirement: The devices and networking equipment needed for the implementation of each solution.
c. Data Handling: This factor involves assessing how each solution manages data, including tasks such as handling training schedules, materials, rules inspection data, and production scheduling data.
d. Cost: Consider the overall expenses associated with each solution, including web development costs, subscription fees, and hardware.
e. Programming Language: The programming languages utilized by each solution, as this can influence development flexibility and maintenance efforts.

No.	**Comparison Entity**	**Alternative Solution 1**	**Alternative Solution 2**	**Alternative Solution 3**
1.	Software	● Interface Design: HTML, CSS, JavaScript ● Backend Development: Python	● Microsoft Excel	● Microsoft Power Apps ● SharePoint ● Power Automate ● Microsoft Visual Studio Code

		● Google Calendar API		
2.	Hardware	● PC ● Laptop ● Networking Equipment (Routers, Switches, etc)	● PC ● Laptop	● Mobile devices (Smartphone) ● Tablets
3.	Data	● Schedule Training ● Training Material	● Rules inspection data ● Production scheduling data	● Rules inspection data ● Production scheduling data
4.	Cost	● Web development Cost	● Web development Cost	● Software Subscription cost
5	Programming Language	● HTML, CSS, JavaScript ● Python	● VBA Excel	● PHP ● Javascript ● Python

3.2.2 Quantitative Solution Selection

Criteria for Evaluation:

1. Development Time (out of 5): This criteria assesses the speed and efficiency the software can be developed.

2. Real-time Data (out of 5): This criteria measures the capability of the software to process and display real-time data.

3. Maintenance/Downtime Reduction (out of 5): This criteria evaluates the software ability to minimize downtime through an effective maintenance process.

4. User-Friendly Interface (out of 5): This criteria assesses the accessibility of the software user interface. A higher rating indicates an interface that is easy to navigate, visually appealing, and enhancing overall user satisfaction and productivity.

5. Quality Assurance Capability (out of 5): This criteria measures the effectiveness of the software quality assurance processes in ensuring reliability, security, and performance.

6. Cost-effectiveness (out of 5): This criteria evaluates the economic efficiency of the software solution.

Criteria	Scale
Development Time	**1.** Development time is slow and inefficient, significantly delaying the software completion. **2.** Development time is below average, causing some delays in software completion. **3.** Average development time, with moderate efficiency in completing the software. **4.** Development time is above average, ensuring timely completion of the software. **5.** Excellent development time, demonstrating high speed and efficiency in completing the software.

Real-time Data	**1.** No capability for processing or displaying real-time data. **2.** Limited capability for real-time data processing, leading to considerable delays. **3.** Moderate capability for real-time data processing, with some delays. **4.** Good capability for real-time data processing, with minimal delays. **5.** Excellent capability for real-time data processing, with rapid and seamless data updates.
Maintenance/Downtime Reduction	**1.** Maintenance processes are ineffective, causing frequent and extended periods of downtime. **2.** There is a limited capacity to reduce downtime, resulting in frequent interruptions. **3.** Maintenance processes have a moderate effect on minimizing downtime, but there is room for improvement. **4.** Maintenance processes are effective in significantly reducing downtime. **5.** Maintenance processes are highly effective, nearly eliminating any downtime.

User-Friendly Interface	**1.** Interface is difficult to navigate, visually unappealing, and hindering user productivity. **2.** Below-average interface, causing some usability issues and dissatisfaction among users. **3.** Average interface, functional but lacking in visual appeal and user satisfaction. **4.** Good user-friendly interface, enhancing user satisfaction and productivity. **5.** Excellent user-friendly interface, easy to navigate, visually appealing, and greatly enhancing user satisfaction and productivity.
Quality Assurance Capability	**1.** Ineffective quality assurance processes, resulting in frequent software failures and security breaches. **2.** Limited quality assurance capability, leading to frequent issues with reliability, security, and performance. **3.** Moderate quality assurance capability, ensuring some reliability, security, and performance. **4.** Good quality assurance capability, ensuring reliability, security, and performance with minimal issues. **5.** Excellent quality assurance capability, ensuring high reliability, security, and performance with virtually no issues.

Cost-effectiveness	**1.** Highly costly solution, surpassing budget limits **2.** Expensive solution, putting strain on the budget and possibly leading to financial challenges. **3.** Moderately priced solution, staying within budget. **4.** Budget-friendly solution, staying within budget limits and offering moderate savings **5.** Extremely cost-effective solution, saving significant resources and costs.

Alternative Solution 1

No.	Quantitative Solution Selection	Score	Reason
1.	Development Time	2	The solution development time is rated as moderate, suggesting that it would take a reasonable amount of time to complete. However, there might be areas where efficiency could be improved to expedite the development process further.
2.	Real-time Data	3	While the solution has some capability for real-time data processing, there are indications that it may not be fully optimized, leading to potential delays in processing real-time information. This suggests a need for refinement in this aspect to ensure timely updates and data accuracy.

3.	Maintenance/Downtime Reduction	3	The solution demonstrates average performance in reducing maintenance and downtime. While it performs adequately in this area, there is room for improvement to streamline processes further and minimize disruptions effectively.
4.	User-Friendly Interface	5	This solution offers a user-friendly interface. Its interface is intuitive, easy to navigate, visually appealing, and enhances overall user satisfaction and productivity.
5.	Quality Assurance Capability	4	The solution exhibits good quality assurance capability, ensuring reliability, security, and performance with minimal issues. However, there might be minor areas where further enhancements could be made to optimize quality assurance processes.

6.	Cost-effectiveness	2	The existing solution lacks sufficient cost-effectiveness, which could strain the budget due to its below-average performance in this aspect.
Total		**19**	Although the solution offers a user-friendly interface and strong quality assurance, it struggles with cost-effectiveness, potentially burdening the budget. Furthermore, its development time and real-time data capabilities are only moderate, indicating areas where enhancements could be beneficial.

Alternative Solution 2

No.	**Quantitative Solution Selection**	**Score**	**Reason**
1.	Development Time	3	The solution development time is rated as average,it would take a reasonable amount of time to complete. This indicates a satisfactory pace of development but leaves room for improvement to enhance efficiency further.
2.	Real-time Data	4	This solution demonstrates good capability for real-time data processing, ensuring timely updates and accurate data representation. Its efficient processing of real-time information contributes to its effectiveness in meeting operational needs.

3.	Maintenance/Downtime Reduction	3	The solution demonstrates average performance in reducing maintenance and downtime. There is potential for improvement in optimizing these processes. Streamlining maintenance procedures and minimizing downtime could further enhance operational efficiency.
4.	User-Friendly Interface	2	The solution interface is rated as below average, potentially impacting user satisfaction and productivity. Improvements in interface design and usability could lead to better user experiences and increased efficiency.
5.	Quality Assurance Capability	3	The solution shows satisfactory quality assurance capability, ensuring reliability, security, and performance to a reasonable extent. There might be areas where quality assurance processes could be strengthened to enhance overall software quality.

6.	Cost-effectiveness	3	The solution demonstrates average cost-effectiveness, falling within budgetary constraints but with limited cost savings.
Total		**18**	The total score is indicating a moderate overall performance. While the solution performs well in real-time data processing and meets quality assurance standards, improvements are needed in areas such as user interface and cost-effectiveness

Alternative Solution 3

No.	Quantitative Solution Selection	Score	Reason
1.	Development Time	3	In this project, the evaluation of the solution development time indicates an average rating. This implies that the pace of development is considered reasonable, striking a balance between efficiency and thoroughness.
2.	Real-time Data	4	This solution demonstrates good capability for real-time data processing, ensuring timely updates and accurate data representation. Its efficient processing of real-time information contributes significantly to its effectiveness in meeting operational needs.

3.	Maintenance/Downtime Reduction	4	The solution performs well in reducing maintenance and downtime, indicating effective processes in place to minimize disruptions. Its streamlined maintenance procedures contribute to enhanced operational efficiency and system reliability.
4.	User-Friendly Interface	5	This solution offers a user-friendly interface, enhancing user satisfaction and productivity.
5.	Quality Assurance Capability	4	The solution demonstrates good quality assurance capability, ensuring reliability, security, and performance with minimal issues.

6.	Cost-effectiveness	4	This solution is very budget-friendly because it mostly uses subscription-based payment methods. This means it saves a lot of money by sticking closely to the existing budget.
Total		**24**	This solution is indicating the overall effectiveness. The user-friendly interface, strong quality assurance, and effective maintenance processes contribute to its appeal. Additionally, the outstanding cost-effectiveness, particularly achieved through the implementation of subscription-based payment models

3.2.3 Solution Selection

Alternative solution 3 The Quality Control Mobile Application is the top choice for improving quality control in manufacturing, scoring a great 24 out of 30 in the quantitative assessment. The greatest strength is in the user-friendly interface this means it is easy for inspectors to use and makes their jobs more efficient. This mobile app helps the inspection process by providing inspectors with a useful tool on their smartphone or their tablets. This allows the inspector to move around the manufacturing site freely while checking and also saving more time and improving the accuracy

One of the benefits is the application design and automated features, which help reduce errors during inspections. By guiding inspectors through the standardized processes and automating tasks like for example data entry and report generation, the app ensures consistency and reliability in quality assessment practices. Moreover, The Quality Control Mobile Application offers significant cost savings and efficiency gains for manufacturing companies. Unlike traditional solutions that require expensive hardware and infrastructure, this app is accessible and affordable. Its subscription based pricing model makes it a cost effective option for businesses of all sizes.

Furthermore, the Quality Control Mobile Application enhances teamwork and communication among inspection teams and management. With features like real-time data sharing and instant notifications, the app makes it easy for inspectors and supervisors to work together effectively. This improved communication ensures that problems are solved quickly and decisions are made promptly. It also strengthens collaboration and encourages everyone to work together to make things better. Overall, the app helps manufacturing companies run more smoothly and improve their processes continuously.

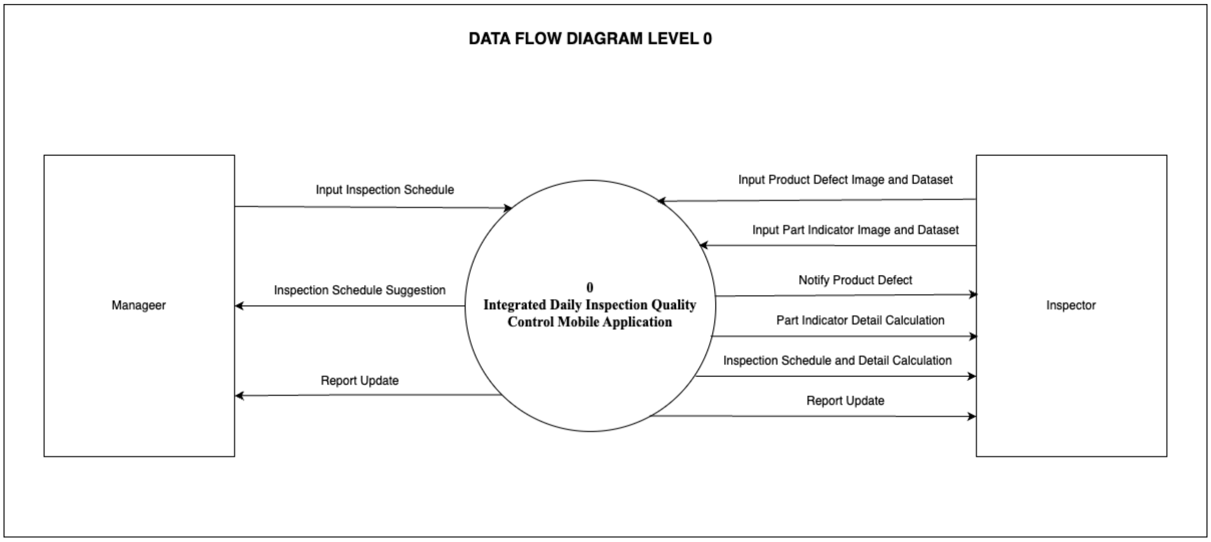
In summary the Quality Control Mobile Application is a changer for QC in manufacturing. Its user friendly interface, streamlined process and cost effectiveness make it an invaluable tool for improving product quality and efficiency in manufacturing environments

3.3 Hierarchical/Iterative Design

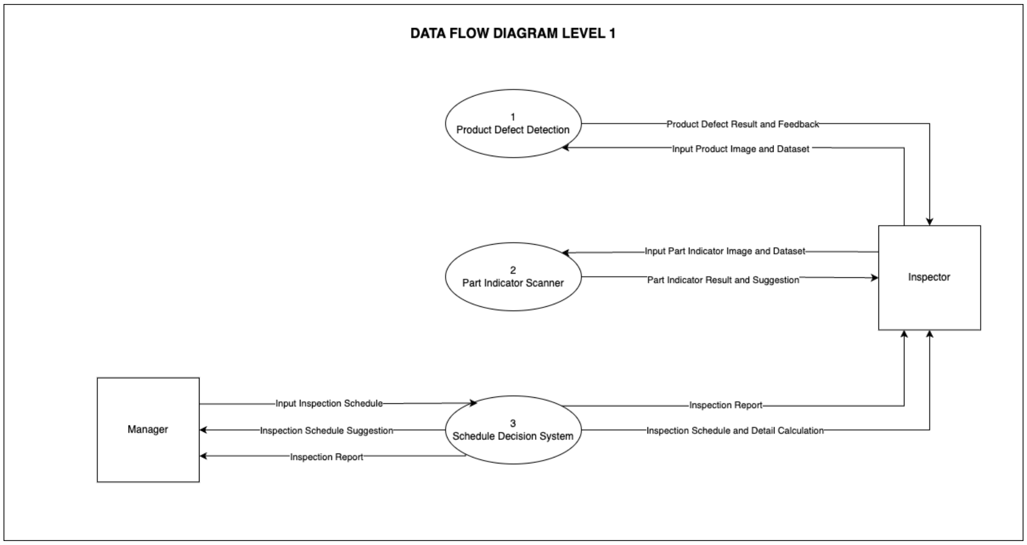
3.3.1 Data Flow Diagram of Defect Detection System, Part Indicator Scanner System, Schedule Decision System.

The methodologies and design of software engineering are crucial in guiding our team to navigate a development process and achieve the project's objective. This section will discuss numerous comprehensive explanations of logic systems and processes.

A. Data Flow Diagram Level 0 (Context Diagram)


DATA FLOW DIAGRAM LEVEL 0
Manageer
Input Inspection Schedule
Inspection Schedule Suggestion
Report Update
0
Integrated Daily Inspection Quality Control Mobile Application
Input Product Defect Image and Dataset
Input Part Indicator Image and Dataset
Notify Product Defect
Part Indicator Detail Calculation
Inspection Schedule and Detail Calculation
Report Update
Inspector

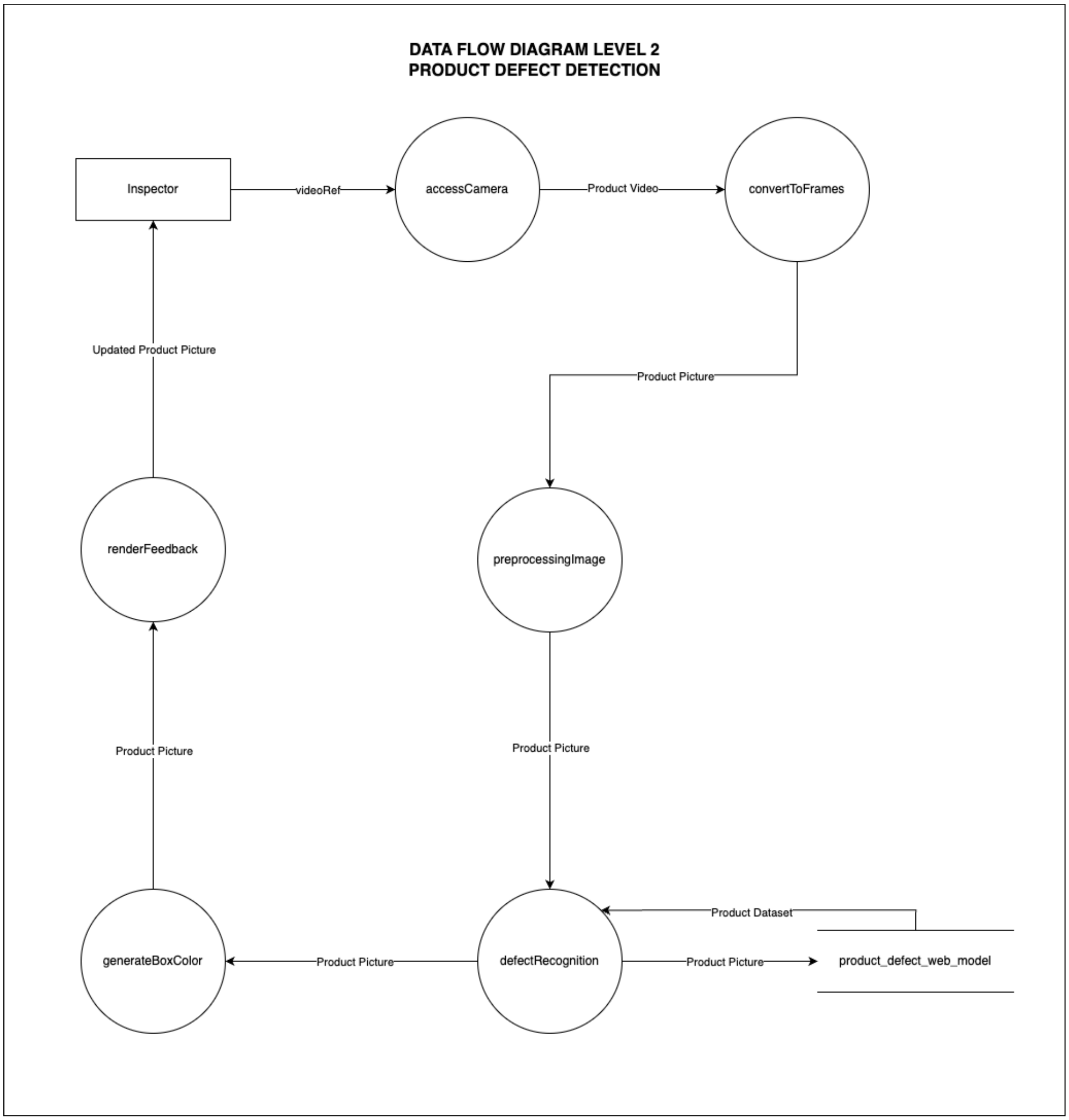
B. Data Flow Diagram Level 1


DATA FLOW DIAGRAM LEVEL 1
1
Product Defect Detection
Product Defect Result and Feedback
Input Product Image and Dataset
2
Part Indicator Scanner
Input Part Indicator Image and Dataset
Part Indicator Result and Suggestion
Inspector
Manager
Input Inspection Schedule
Inspection Schedule Suggestion
Inspection Report
3
Schedule Decision System
Inspection Report
Inspection Schedule and Detail Calculation

This is our data flow diagram Level 0, which describes every procedure inside our modules and functions. It provides a thorough rundown of all three modules' connections. The Decision Detection System, the first module, processes the scanned object input that the Inspector provides, indicates any defects, and then sends the processed data back to the Inspector. In the second module, Part Indicator Scanner System receives a scanned object from the inspector and gives the suggestion result back to the Inspector as the output. The third module, the Schedule Decision System, involves inputs from both user types which are the Manager and the Inspector. Manager can manage the schedule and receive the reporting, but the Inspector can choose the schedule as an input and receive details along with the decision result as an output.

1. **Product Defect Detection**

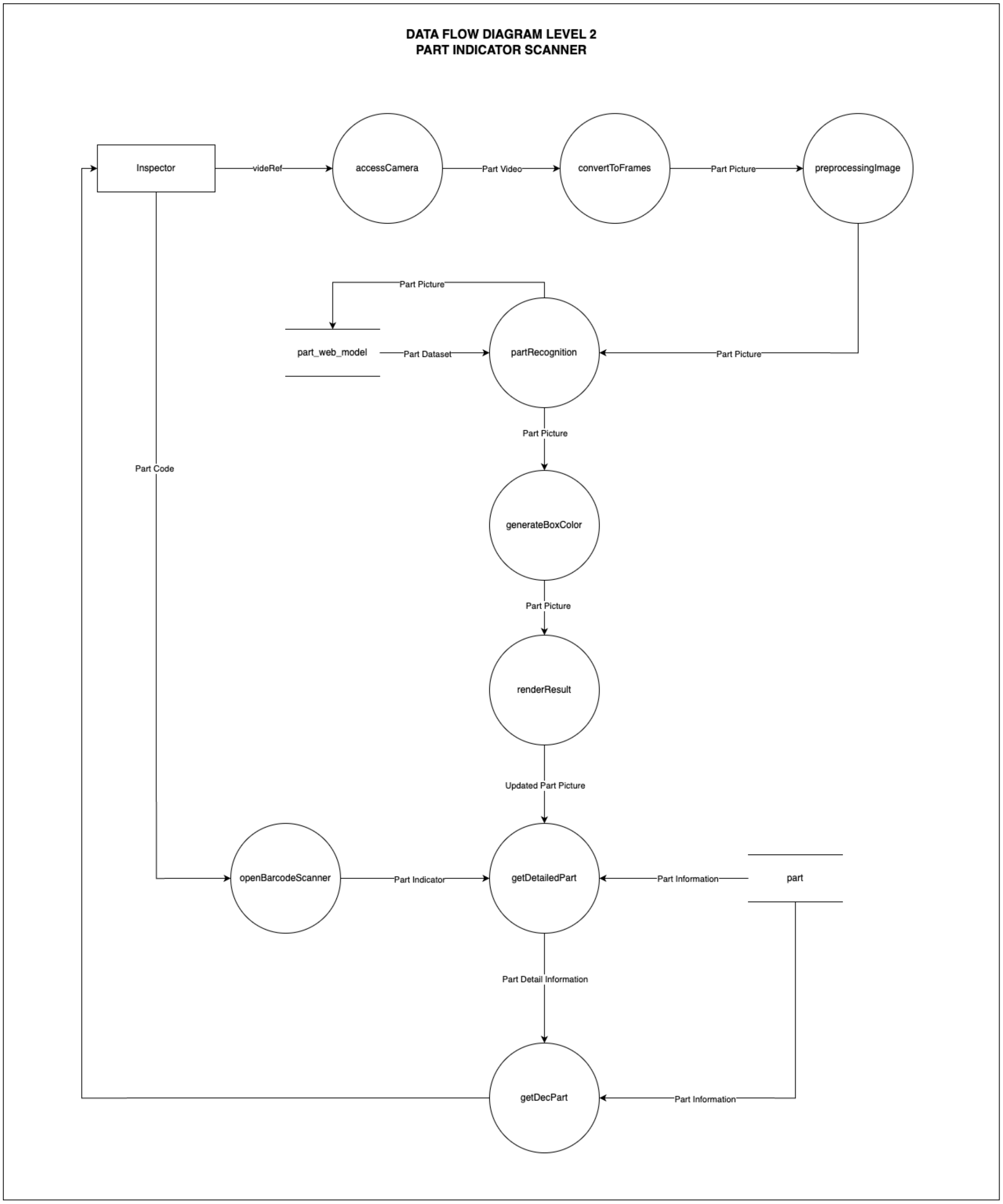
- Data Flow Diagram Level 2



In the Data Flow Diagram above, it explains the level 2 Data Flow Diagram of Defect Detection System. It describes how the module initially gets an input of a scanned object from the Inspector. The system then goes through convertToFrames on the scanned object (Product Video) , makes it the input as image and moves on to preprocessing. After the properties have been configured, the system checks it for defects. The Inspector receives defect feedback as the final outcome after the entire process is finished.

2. Part Indicator Scanner

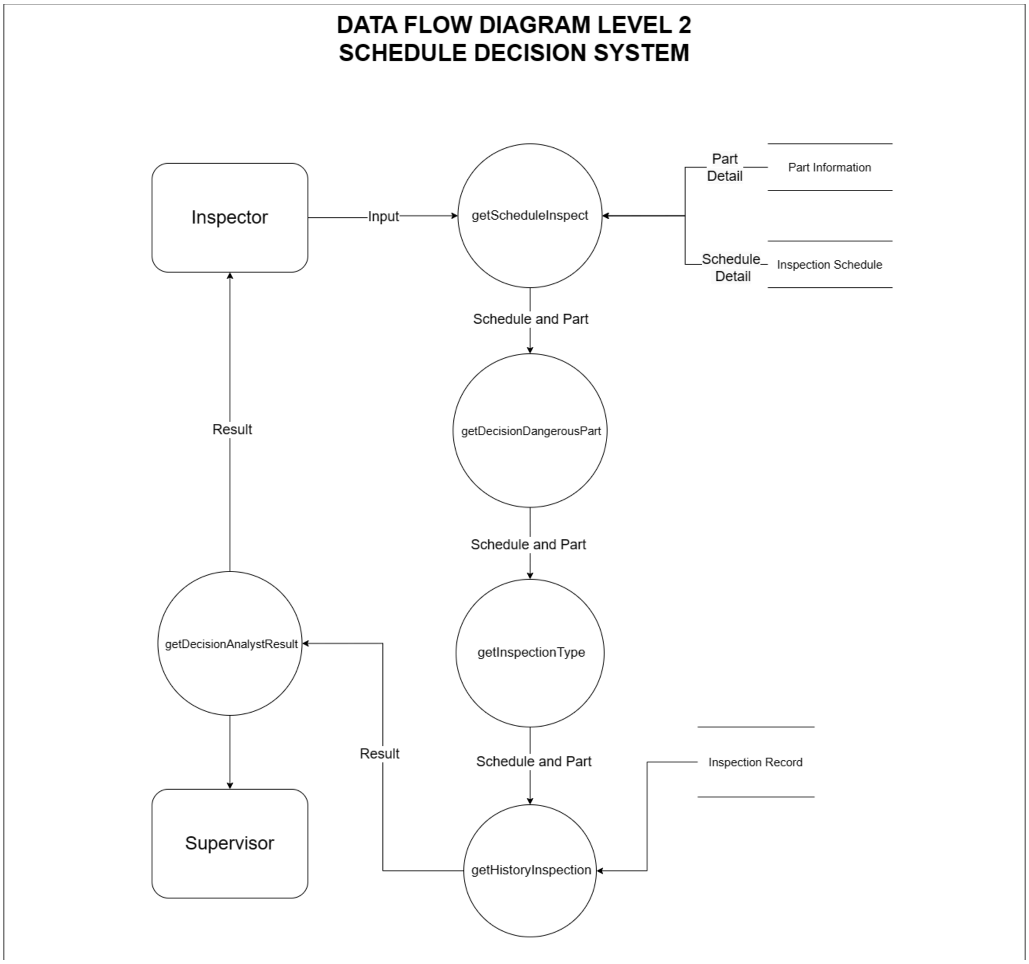
- Data Flow Diagram Level 2



The level 2 Data Flow Diagram of the Part Indicator Scanner System is explained in the data flow diagram above. It describes how the module's initial process flow works. An input from the inspector is sent as a scanned part or scanned object (Part Video). The system receives it, converts the video into the frames (image), and then performs preprocessingImage. After the machine learning process has identified the properties and the part, the system will obtain the associated data from the identified part, otherwise the Inspector could directly scan code to detect the part indicator. Then the system receives the dangerous information about the part. Finally, all finished processed data will be sent back to the Inspector.

3. Schedule Decision System

- Data Flow Diagram Level 2



The provided diagram above is the level 2 Data Flow Diagram for the Schedule Decision System. The level 1 Data Flow Diagram for the Schedule Decision System is shown in the diagram above. The Inspector in this system inputs selected schedules, which start all necessary processes. Based on the selected schedule, the system then processes the data from the inspection schedule and part information. After that, a decision-making process takes place, involving data analysis, investigation into potential options, and application of best practices for analyzing the current product. The Inspector then receives the feedback as the output. Schedules can be submitted to the system by the Manager. The system suggests scheduling inputs to the Manager in addition to adding these schedules directly by utilizing its decision-making process. Additionally, the Manager can receive performance results as an output, which improves the process of making managerial decisions.

3.3.2 Global Variable / Message Passing Interface

Global Variable, Parameter passing and encapsulation concepts are used in this Integrated Daily Inspection Quality Control Mobile Application to transfer information between modules, It is the process of exchanging data between two different modules in the system. Software design and object-oriented programming (OOP) are related concepts that involve encapsulation and parameter passing.

1. Defect Detection System

Encapsulation is used in the Defect Detection System to protect internal variables and business processes. This module's global variables serve as a storage for important data that must be shared between various components. Defect-related information could be contained in a class so that only authorized module components could access it. Message passing are used to enable real-time communication between the pre-trained model and the defect detection algorithm, below is the flow :

a. Processing Dataset (Sending Dataset)
   The Dataset contains a pre-trained model of Product Defect and ready to send to the next step.
b. Process Receiving Dataset (Receive Dataset)
   The Defect Detection System can do detection with the received Dataset.

2. Part Indicator Scanner System

Global variables encapsulation makes sure that important information about scanned parts is safely kept and accessible in the Part Indicator Scanner System. This ends accidental disruption of important data. Message passing is used to enable real-time communication between the pre-trained model and the defect detection algorithm, below is the flow :

a. Processing Dataset (Sending Dataset)

   The Dataset contains a pre-trained model of Part Indicator and ready to send to the next step.

b. Process Receiving Dataset (Receive Dataset)

   The Part Indicator Scanner System can do detection with the received Dataset.

3. Schedule Decision System

Encapsulation is used by the Schedule Decision System to protect the integrity of global variables that hold scheduling-related data and scheduling algorithms. For this module to communicate with the Defect Detection System and the Part Indicator Scanner System, message passing is crucial. In order to ensure flexibility in response to real-time changes in the inspection procedure, the system also used parameter passing functionality to provide the result for scheduling decisions.

a. Processing Dataset (Sending Dataset)

   The Dataset contains a set of required inputs and matrix parameters and is ready to send to the next step.

b. Process Receiving Dataset (Receive Dataset)

   The Schedule Decision System can do decision making with the received Dataset.

3.3.3 Waterfall Methodology to Implement Steps in Software Engineering Design

Our team has decided to implement the Waterfall Methodology to our capstone project. A classic, straight approach to software development, the waterfall model consists of step-by-step flow through separate stages. This decision is supported by the needs of our project, which has well-defined and stable requirements, and where a structured, step-by-step process is considered beneficial. These are the steps that we implement :

1. Requirements Analysis:

   During the first stage, we collect and record all of the specific software and business process requirements. Before moving on to design, this step ensures a thorough understanding of end-user needs especially in the Quality Control division.

2. System Design:

   Afterwards, we develop a high-level system design that describes the general architecture, parts, and software system modules. Additionally, each component's or module's further details are covered in this phase. For guidance during the implementation stage, comprehensive specifications are created, including flowcharts and interface details. A clear development roadmap is provided by this step.

3. Implementation:

   Our team starts developing the software and modules by coding using the comprehensive design specifications as a guide, following acknowledged coding standards and best practices. To ensure functionality and stability, unit tests aim to validate individual components.

4. Testing:

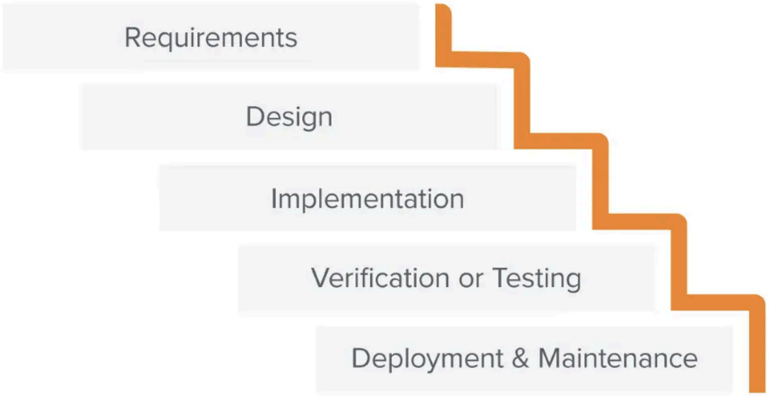
   Comprehensive testing is carried out step-by-step and involves system, integration, and unit testing. Before going on to the next stage, identified bugs and weaknesses are fixed to guarantee that the implemented system satisfies the requirements.

5. Deployment:

   We move on to deployment with a fully tested and completed software product. Documentation and user guides are created to ensure an insightful and helpful release for end users to use the product.

6. Maintenance:

   After deployment, we actively resolve bugs and reported issues while also making the required updates. The software's continued functionality is ensured during the maintenance phase, which also offers a way to integrate user feedback to our development.



3.3.4 Detail of Component and Library Used

a. **Hardware:**

- **Mobile Tab/Phone**

  A mobile tablet or a mobile phone are the main pieces of hardware needed to run the application. These devices are meant to show the dashboard for the application and make scanning functionalities easier so that features like part indicator scanners and defect detection systems can be enabled.

b. **Software:**

- **Web Browser**

  Web browsers are software programs that are used to view and access camera detection features, such as part indicator scanner systems and product defect detection, because these features are web based.
- **Visual Studio Code**

  VS Code is a top-notch code editor that developers can use to enhance the quality and efficiency of software development, as well as make integration with other technologies easier such as machine learning. It is one of the best tools available for this purpose.
- **Microsoft Power Apps**

  With Microsoft Power Apps, we can easily create custom mobile applications using a robust low-code development platform. It supports our team to develop and integrate with Microsoft SQL Server as well at a time.
- **Microsoft SQL Server**

Comprehensive relational database management system (RDBMS) is provided by Microsoft SQL Server. We use it because Microsoft SQL Server offers robustness, safety and scalability. Without sacrificing dependability or performance, it can manage massive data volumes. Our mobile applications built on top of SQL Server can expand easily as the volume of data in Quality Control grows over time

c. Libraries:

- **YOLOv8**

  One of the best algorithms for real-time object detection at the moment is called YOLO (You Only Look Once). as a real-time object detection model that enhances our two primary features. We use YOLOv8 in our project to support the fast-paced environment of quality control since it is well-known for its ability to generate accurate and timely object detection predictions.

- **React.js**

  React.js is an open-source front-end JavaScript library. It makes it easier for us to create websites and apps. We use it because it makes writing JavaScript code more organized and cleaner. With React we can easily implement object detection modules.

- **Tensorflow.js**

  An open-source machine learning library called Tensorflow.js is used to create and train machine learning models, including YOLO and other object detection models. TensorFlow is used in the project to create, train, and assess object detection models.

d. Connection:

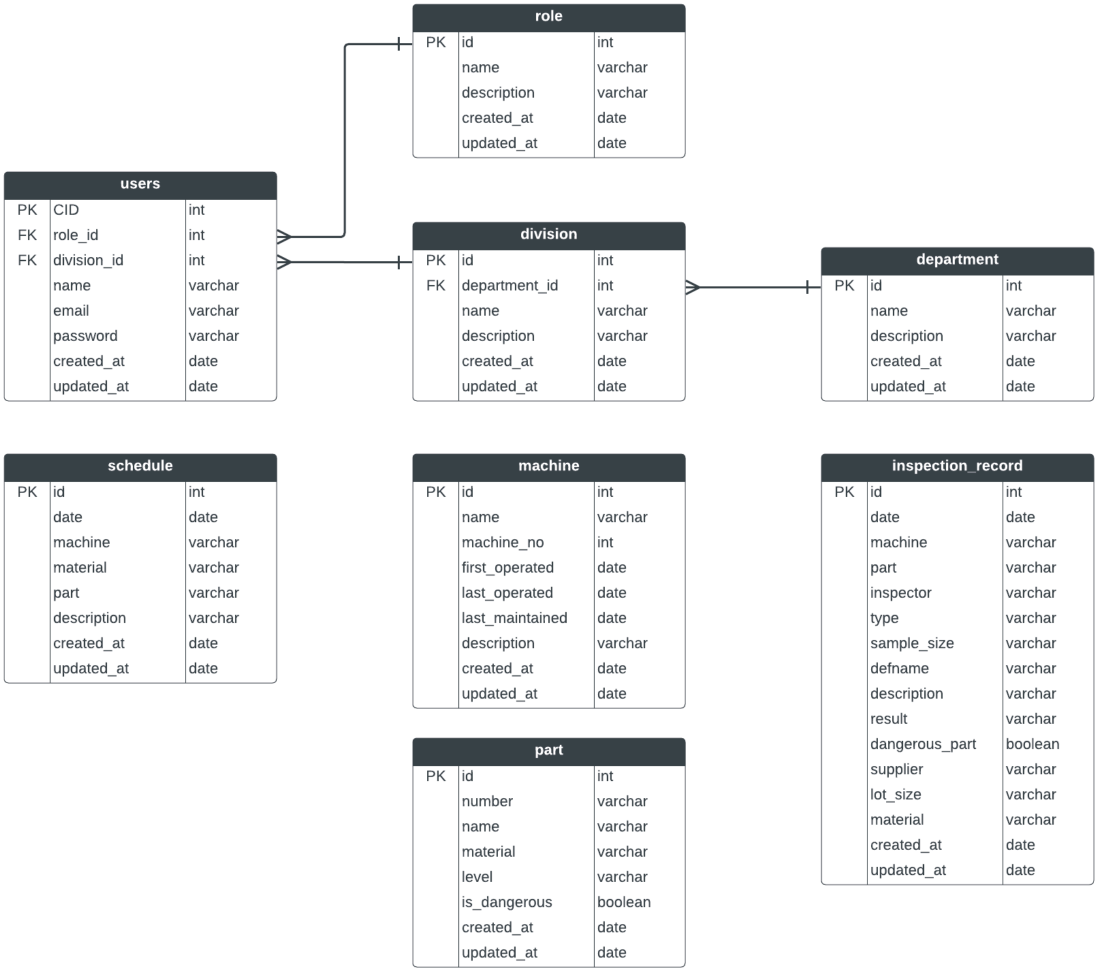
- **Wi-Fi Connection**

  With Wi-Fi, devices can connect wirelessly. In our project, a local server hosts the application that can be accessed via Wi-Fi. Only authorized

Quality Control or company personnel are permitted to use this private application. Wi-Fi facilitates safe access to the application by staff members from any location in the office, which accelerates workflow management and communication.

3.3.5 UML (Structure Diagrams, Behavioral Diagrams)

A. Entity Relationship Diagram



role

PK	id	int
	name	varchar
	description	varchar
	created_at	date
	updated_at	date

users

PK	CID	int
FK	role_id	int
FK	division_id	int
	name	varchar
	email	varchar
	password	varchar
	created_at	date
	updated_at	date

division

PK	id	int
FK	department_id	int
	name	varchar
	description	varchar
	created_at	date
	updated_at	date

department

PK	id	int
	name	varchar
	description	varchar
	created_at	date
	updated_at	date

schedule

PK	id	int
	date	date
	machine	varchar
	material	varchar
	part	varchar
	description	varchar
	created_at	date
	updated_at	date

machine

PK	id	int
	name	varchar
	machine_no	int
	first_operated	date
	last_operated	date
	last_maintained	date
	description	varchar
	created_at	date
	updated_at	date

part

PK	id	int
	number	varchar
	name	varchar
	material	varchar
	level	varchar
	is_dangerous	boolean
	created_at	date
	updated_at	date

inspection_record

PK	id	int
	date	date
	machine	varchar
	part	varchar
	inspector	varchar
	type	varchar
	sample_size	varchar
	defname	varchar
	description	varchar
	result	varchar
	dangerous_part	boolean
	supplier	varchar
	lot_size	varchar
	material	varchar
	created_at	date
	updated_at	date

Three modules/functions are represented by the Entity Relationship Diagram (ERD) above, which has eleven relational tables in total. These tables' relationships range from one-to-many and one-to-one connections. It supports the data storing of the project and supports the system operation in the production environment. Below are the details of each relationship between the tables :

1. Table role and table users
   We use one to many relationships from table role to table users indicating that a role can have more than one user.

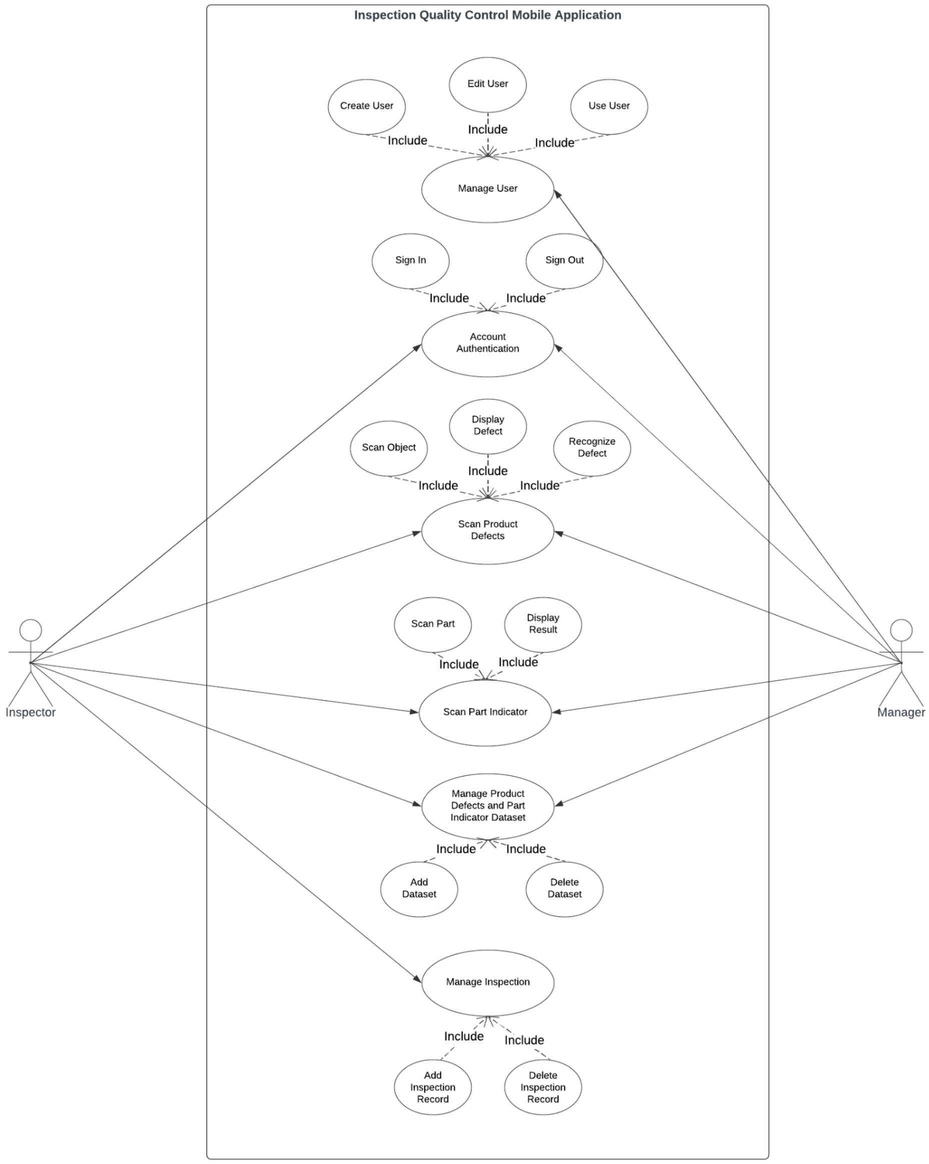
2. Table division and table department

We use one to many relationships from table department to table division indicating that a department can have multiple divisions.

3. Table division and table users

   We use one to many relationships from table division to table users indicating that one division can contain multiple users in it.

   B. Use Case Diagram



The three modules of our system's use case are represented in the diagram above. It defines the operational dynamics and user relationships, explaining how the system works. It covers all aspects of the system such as :

1. Manage User
   It explains how the user of the system can interact with the user data.

2. Account Authentication
   It explains how the user can interact with the authentication security.

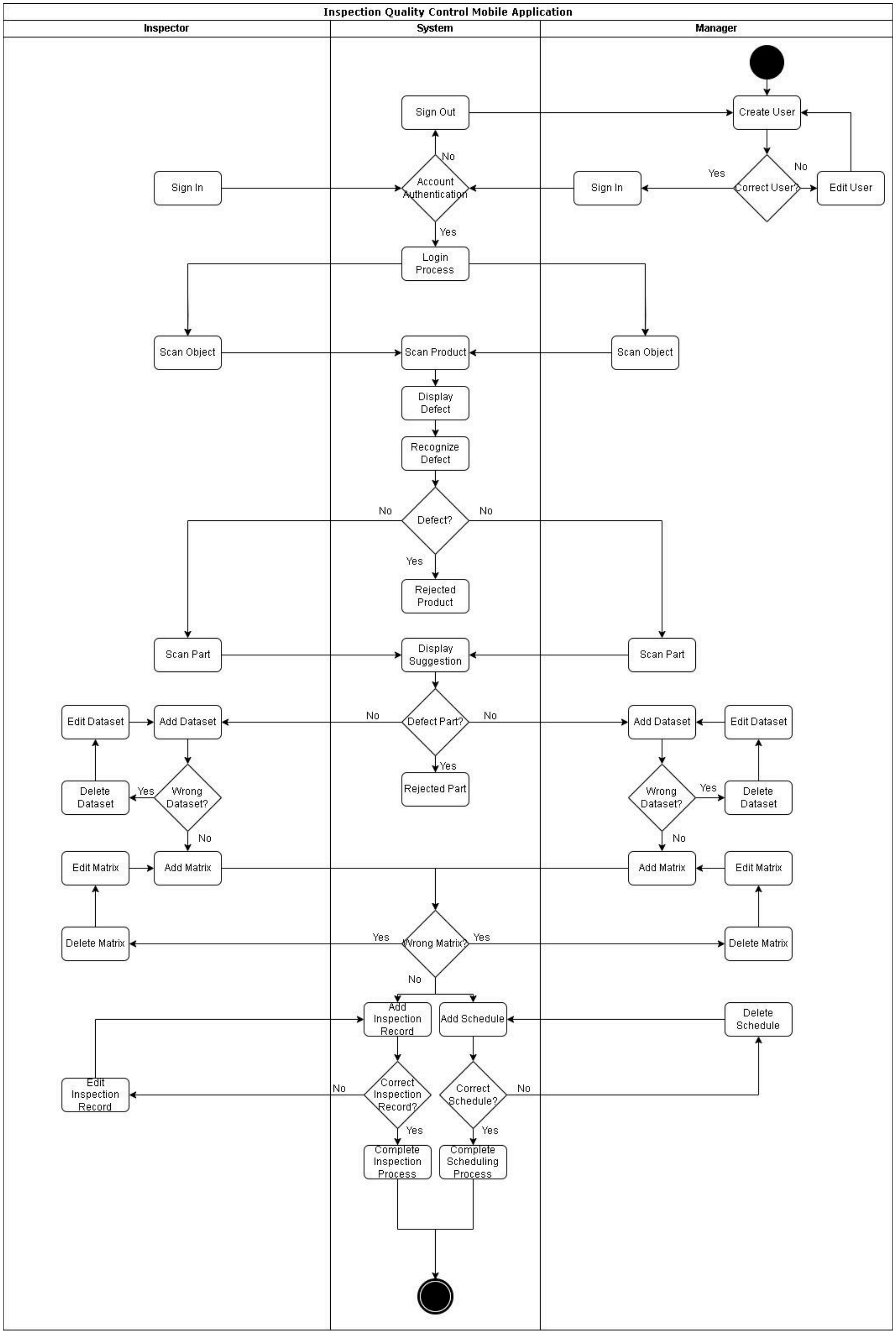
3. Scan Product Defects
   It explains how the user can interact with the product defects system.

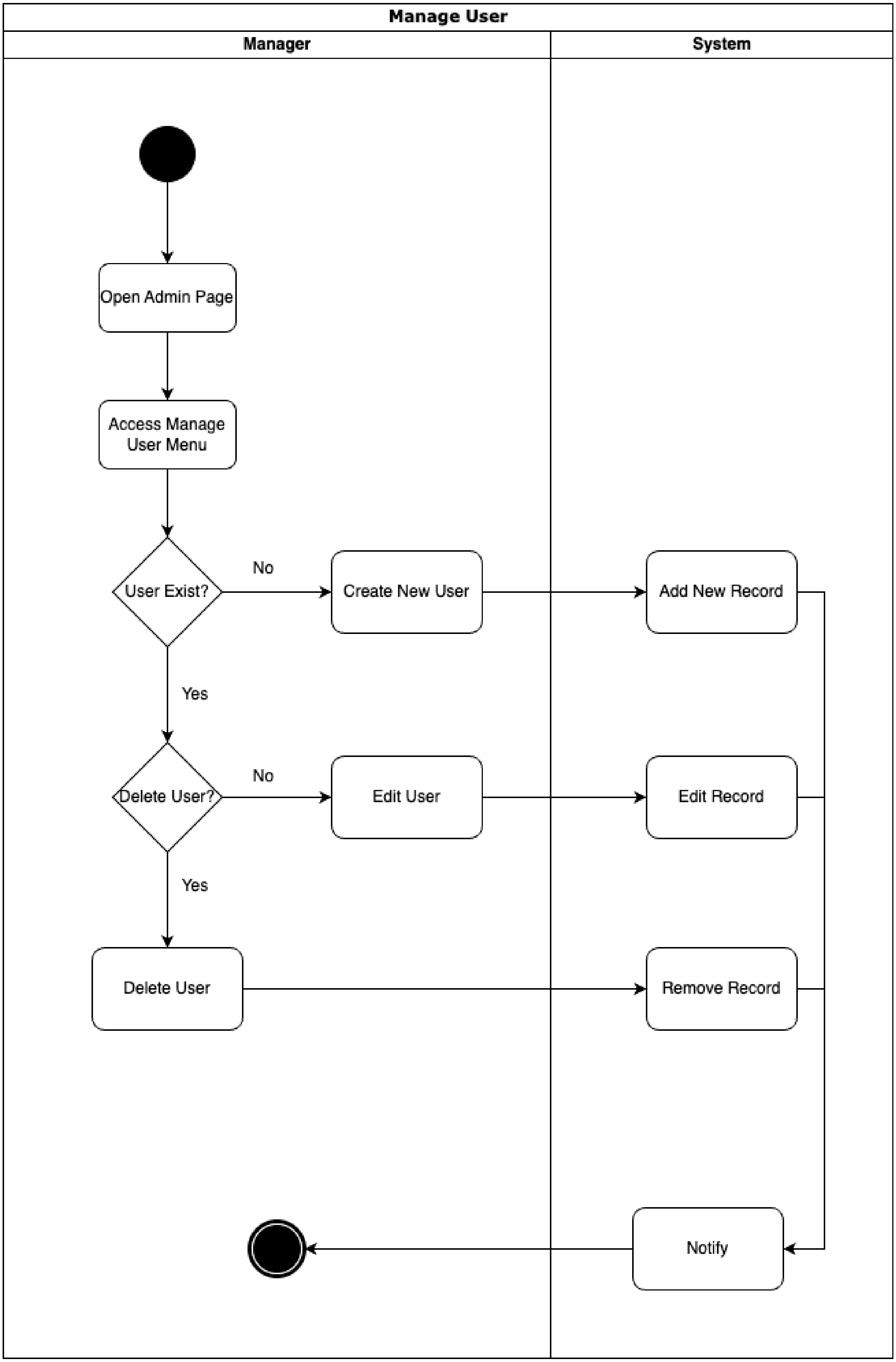
4. Scan Part Indicator
   It explains how the user can interact with the part indicator system.

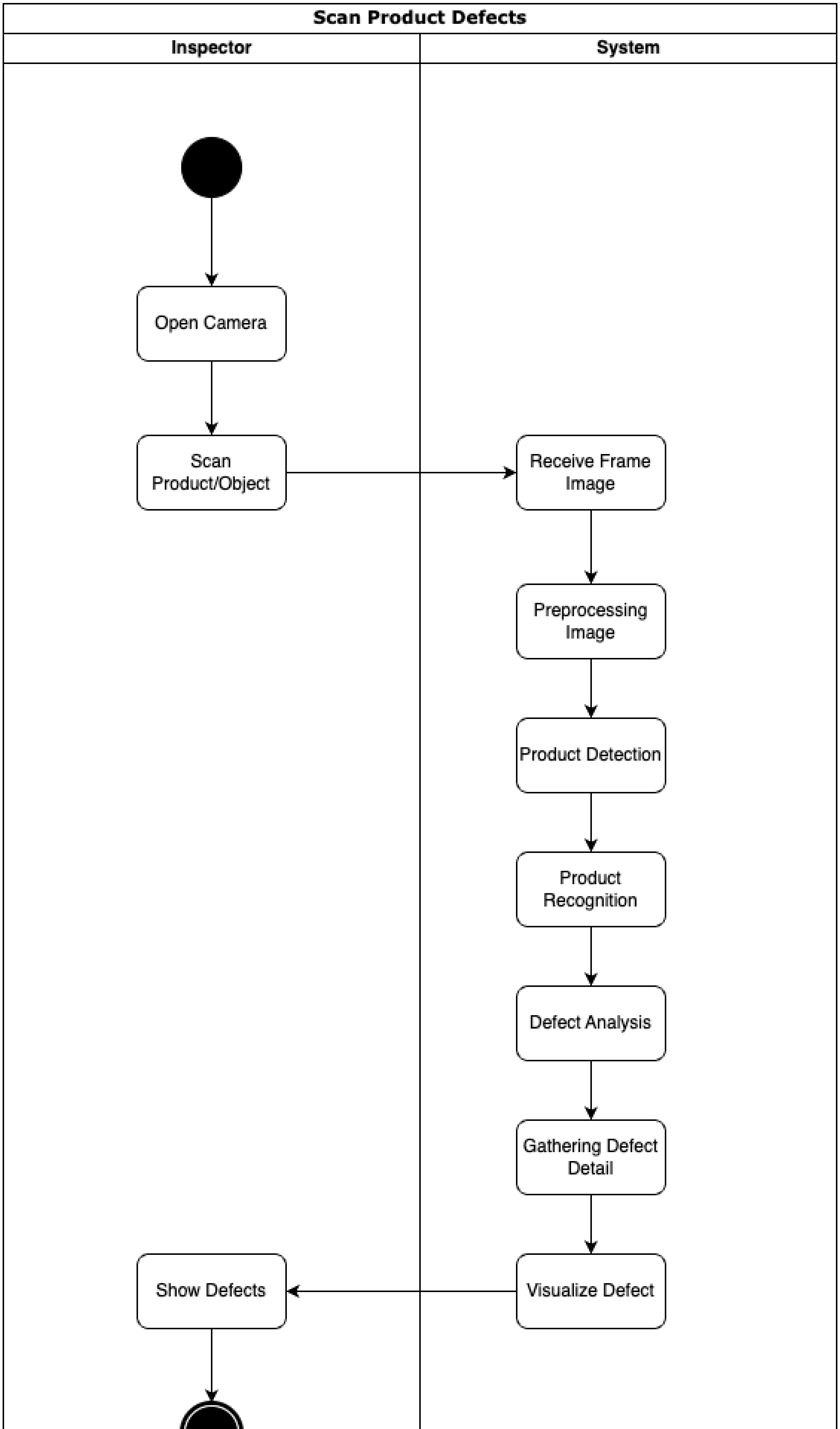
5. Manage Product Defects and Part Indicator Dataset
   It explains how the user can manage with the dataset that will be used for machine learning algorithms to support the modules.

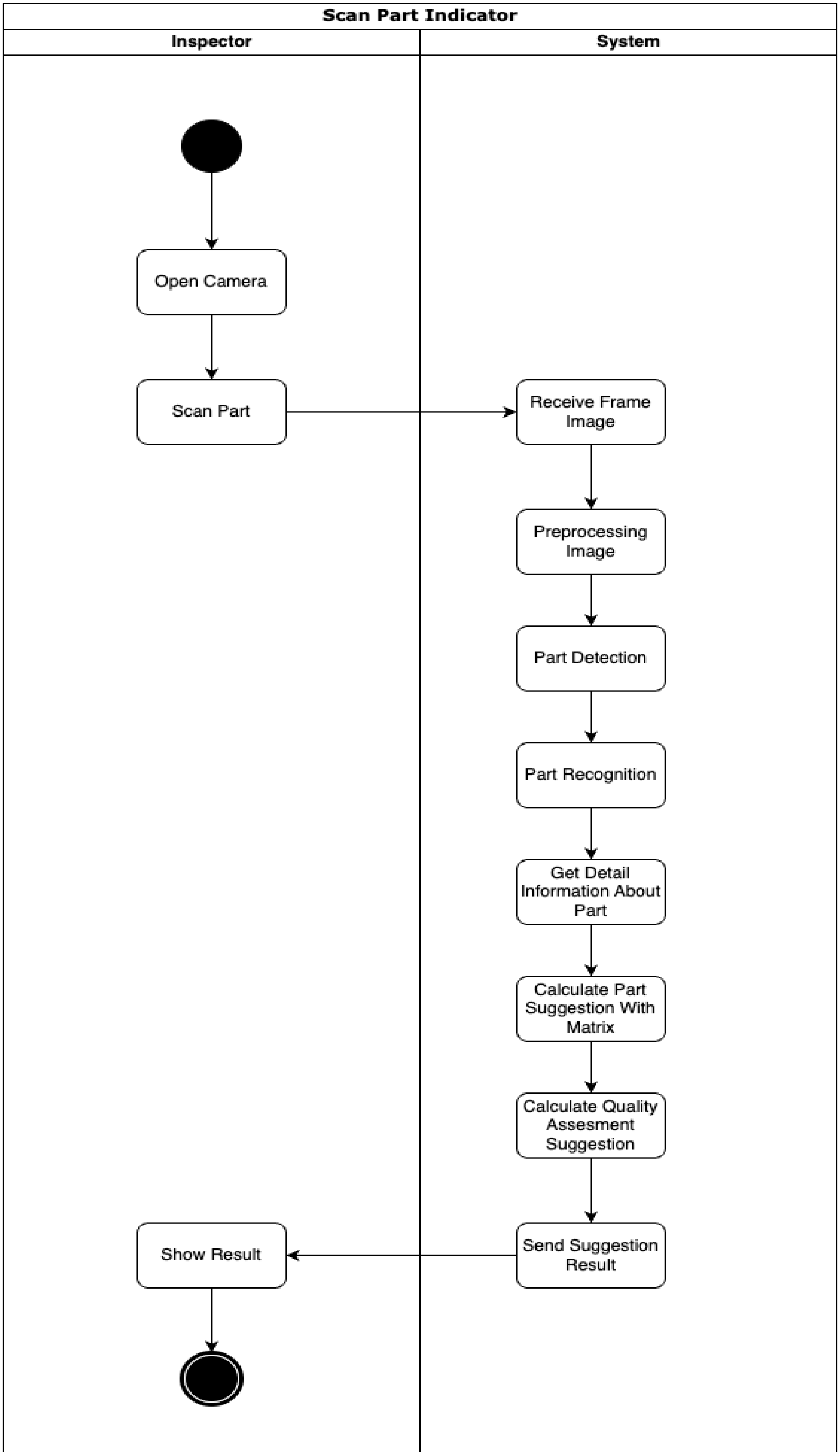
6. Manage Inspection
   It explains how the user can manage and interact with the inspection records.

C. Activity Diagram


Inspection Quality Control Mobile Application
Inspector
System
Manager
Sign Out
Create User
No
Yes
No
Sign In
Account Authentication
Sign In
Correct User?
Edit User
Yes
Login Process
Scan Object
Scan Product
Scan Object
Display Defect
Recognize Defect
No
No
Defect?
Yes
Rejected Product
Scan Part
Display Suggestion
Scan Part
Edit Dataset
Add Dataset
No
No
Defect Part?
Add Dataset
Edit Dataset
Yes
Rejected Part
Delete Dataset
Yes
Wrong Dataset?
Wrong Dataset?
Yes
Delete Dataset
No
No
Edit Matrix
Add Matrix
Add Matrix
Edit Matrix
Yes
Yes
Delete Matrix
Wrong Matrix?
Delete Matrix
No
Add Inspection Record
Add Schedule
Delete Schedule
Edit Inspection Record
No
Correct Inspection Record?
Correct Schedule?
No
Yes
Yes
Complete Inspection Process
Complete Scheduling Process


Manage User
Manager
System
Open Admin Page
Access Manage User Menu
User Exist?
No
Create New User
Add New Record
Yes
Delete User?
No
Edit User
Edit Record
Yes
Delete User
Remove Record
Notify


Scan Product Defects
Inspector
System
Open Camera
Scan Product/Object
Receive Frame Image
Preprocessing Image
Product Detection
Product Recognition
Defect Analysis
Gathering Defect Detail
Visualize Defect
Show Defects


Scan Part Indicator
Inspector
System
Open Camera
Scan Part
Receive Frame Image
Preprocessing Image
Part Detection
Part Recognition
Get Detail Information About Part
Calculate Part Suggestion With Matrix
Calculate Quality Assesment Suggestion
Send Suggestion Result
Show Result

3.4 Verification Demonstration and Proof of Design Process

No	Scenario	Result	Conclusion
1	An inspector opens the mobile app to begin their daily inspection rounds.	The inspector inputs inspection reports directly, ensuring accuracy and eliminating human error.	The integration of Quality Control Mobile Application using Microsoft Power Apps and Laravel Version 10 enhances efficiency in scheduling and decision-making, ultimately improving overall quality control processes.
2	Using the automatic scanner feature, an inspector scans a product on the manufacturing floor to gather information such as material composition and part type.	Real-time inspection data help in workflow planning, while machine learning categorization improves inventory management, leading to optimized operations.	The application's user-friendly interface and flexible navigation enhance efficiency in data collection, contributing to improved workflow management.
3	An inspector utilizes the camera-based defect detection	The module highlights faulty areas with high precision, allowing	Automation of defect detection tasks improves the speed and accuracy of

	module to identify any faults in the product quickly.	for immediate identification and rectification of defects, thus ensuring high product quality.	inspections, ultimately enhancing overall product quality control.
4	During an inspection, the inspector encounters a complex issue requiring collaboration with colleagues.	The inspector uses the app to share real-time inspection data and collaborate with team members, facilitating swift problem-solving and decision-making.	The app's collaborative features improve communication and coordination among team members, leading to more effective problem resolution.
5	After finishing their inspection rounds, an inspector sends out the inspection report via the mobile app.	The report is sent securely with SSL & HTTPS	The app's strong security measures safeguard the confidential inspection data, ensuring it complies with data protection regulations and remains protected from unauthorized access.

3.5 Standards Used

Standards are essentials that offer guidelines, requirements, and best practices for different aspects of a project. They are essential to maintaining quality, safety, compatibility, and consistency throughout the various stages of development and implementation. The standards we applied to our project are listed below:

Design Modeling Standards

- UML Version 2

Software Development Life Cycle Standards

- Waterfall Method

Technologies

- Microsoft Power Apps
- Laravel 10.x
- Microsoft SQL Server
- Microsoft Sharepoint
- Tensorflow.js
- YOLOv8

Data Formats

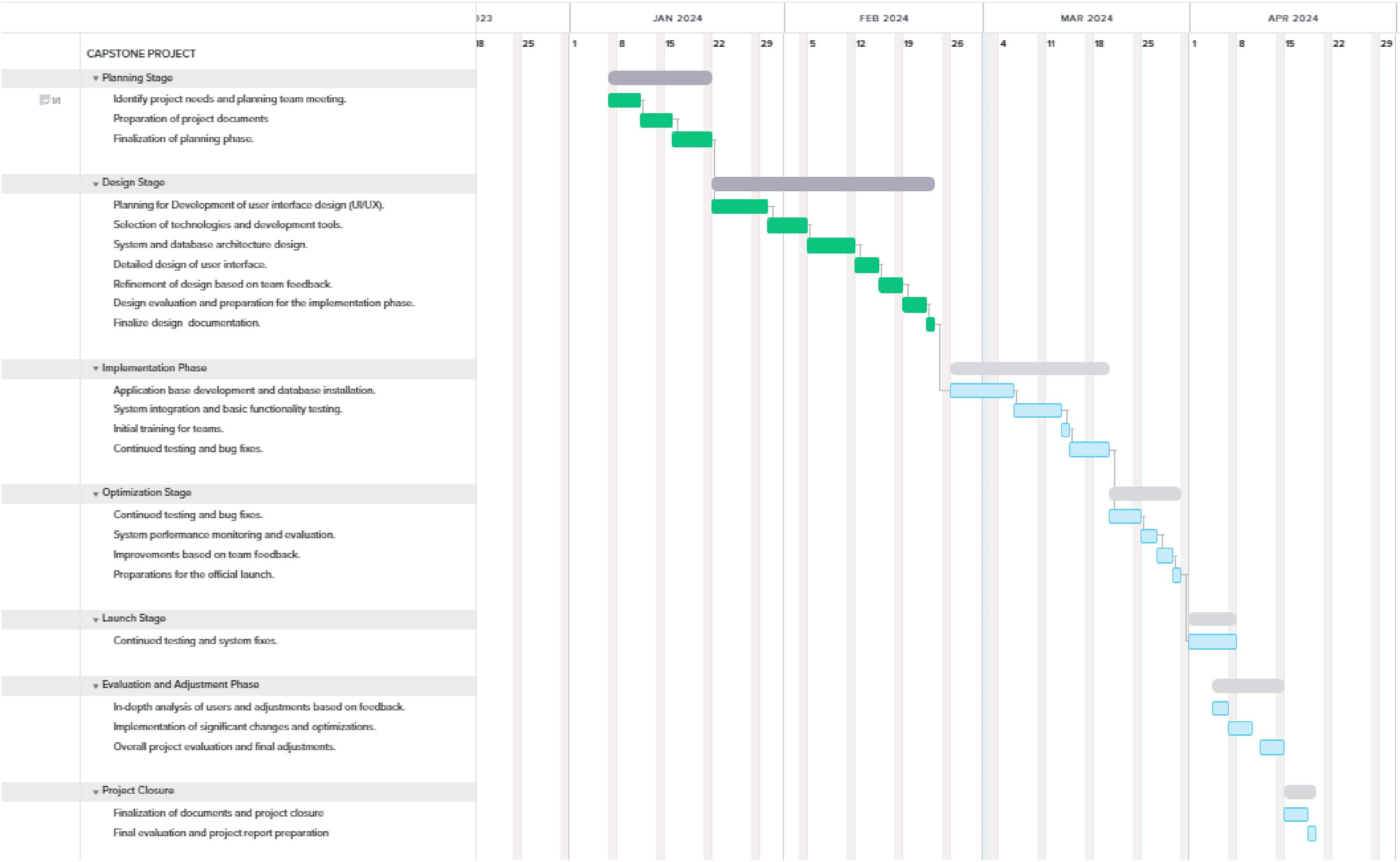
- Javascript Object Notation (JSON)

Security

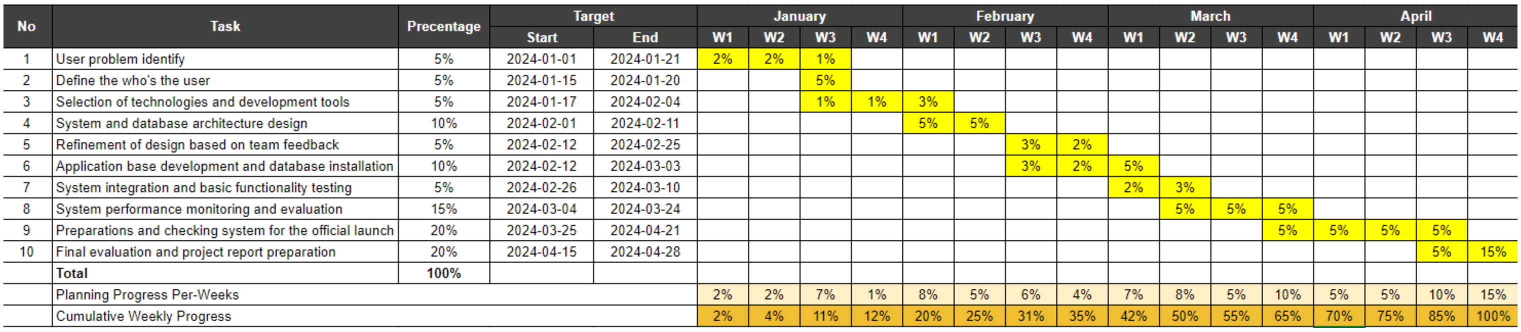
- SSL & HTTPS
- Data Hashing
- Data Encryption

3.6 Implementation and Testing Plans

3.6.1 Gantt Chart



3.6.2 S-Chart



No	Task	Precentage	Target		January				February				March				April			
			Start	End	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
1	User problem identify	5%	2024-01-01	2024-01-21	2%	2%	1%													
2	Define the who's the user	5%	2024-01-15	2024-01-20			5%													
3	Selection of technologies and development tools	5%	2024-01-17	2024-02-04			1%	1%	3%											
4	System and database architecture design	10%	2024-02-01	2024-02-11					5%	5%										
5	Refinement of design based on team feedback	5%	2024-02-12	2024-02-25							3%	2%								
6	Application base development and database installation	10%	2024-02-12	2024-03-03							3%	2%	5%							
7	System integration and basic functionality testing	5%	2024-02-26	2024-03-10									2%	3%						
8	System performance monitoring and evaluation	15%	2024-03-04	2024-03-24										5%	5%	5%				
9	Preparations and checking system for the official launch	20%	2024-03-25	2024-04-21												5%	5%	5%	5%	
10	Final evaluation and project report preparation	20%	2024-04-15	2024-04-28															5%	15%
	Total	**100%**																		
	Planning Progress Per-Weeks				2%	2%	7%	1%	8%	5%	6%	4%	7%	8%	5%	10%	5%	5%	10%	15%
	Cumulative Weekly Progress				2%	4%	11%	12%	20%	25%	31%	35%	42%	50%	55%	65%	70%	75%	85%	100%