King Fahad University of Petroleum and Minerals
COE384
Engineering Design Document (EDD)

Project Name:	Smart Fruit Grading System		
Group Number:	1		
Date:	21/12/2023	**Release:**	Final
Student 1 Name:	Raad Badr Al-Utaibi	**Signature:**	X
Student 1 ID:	201972010		
Author 2 Name:	Basil Alashqar	**Signature:**	X
Student 2 ID:	202045700		
Author 3 Name:	Nussair Hroub	**Signature:**	X
Student 3 ID:	201972210		
Author 4 Name:	Omar Almutairi	**Signature:**	X
Student 4 ID:	201917610		
Document Number:	T231-COE384-EDD-002		

Revision History

Revision Date	Previous Revision Date	Summary of Changes	Changes Marked
30/11/2023	---------	The entire document	Yes
21/12/2023	30/11/2023	The entire document	Yes

Approvals

This document requires the following approvals. A signed copy should be placed in the project files.

Name	Signature	Title	Date of Issue	Version
Dr. Akram		Professor	21/12/2023	002

Distribution

This document has been distributed to:

Name	Title	Date of Issue	Version
Dr. Akram	Professor	21/12/2023	002

Contents

1. Executive Summary of Project Charter

Using the project management plan (T231-COE 384-PMP-002) that extends the business case (T231-COE 384-BC-003), we are continuing to discuss the project of a Smart Fruit Grading System. The Smart Fruit Grading System (SFGS) represents the vanguard of agricultural innovation, poised to catalyze a transformative shift in the traditional paradigm of fruit grading processes. At its heart lies a meticulously engineered hardware architecture, where a central microcontroller orchestrates a symphony of cutting-edge components—cameras, gates, and conveyors. This intricate dance begins with specialized cameras, including an X-ray variant, capturing nuanced images of fruits positioned on a transparent conveyor belt. This visual data is then transmitted to the microcontroller, which, empowered by a Fine-Tuned DenseNet-201 AI model, assumes the role of a discerning mind. Its task: categorize fruits into high, medium, low, and inedible grades. The microcontroller executes this cognitive feat with finesse, guiding the graded fruits through output gates to their predestined lines, thereby streamlining the entire grading process with elegance and efficiency.

On the software front, the SFGS operates seamlessly on the stalwart and versatile Ubuntu 22.04 LTS operating system. The software architecture is an exemplar of engineering finesse, featuring a meticulously crafted hierarchy of components—the User Interface (UI), Reset, AI Model (Fine-Tuned DenseNet-201), Motion Detection Algorithm, and Camera Interface. This sophisticated and layered structure serves not only to enhance the system's operational efficiency but also lays a robust foundation for facile upgrades and maintenance, ensuring the SFGS's perpetual adaptability to the ever-evolving landscape of technological advancements.

Critical engineering decisions underpin the selection of the Fine-Tuned DenseNet-201 AI model, renowned for its impeccable track record in image classification accuracy. The strategic integration of third-party components showcases a forward-thinking approach, leveraging established and optimized solutions within the expansive realm of artificial intelligence. The adoption of a layered software structure underscores a commitment to adaptability, affording the flexibility necessary for the seamless integration of future software enhancements and technological advancements.

In concert with these engineering considerations, the SFGS seamlessly integrates key third-party software components to augment its capabilities. Ubuntu, standing steadfast in version 22.04 LTS, assumes the mantle of a robust foundation for hosting and executing the desktop application and AI model components. Python, in its dynamic iteration 3.8.15, assumes the mantle of the programming language, infusing versatility into various components of the fruit grading and sorting system. PyTorch, in its sophisticated version 2.0.1, emerges as the preferred deep learning framework for implementing and deploying the DenseNet-201 model. NumPy, in version 1.24.3, finds its niche as a PyTorch dependency, providing fundamental support for multi-dimensional arrays and mathematical operations. OpenCV, in its advanced version 4.8.0, seamlessly integrates into the system, contributing its robust capabilities to image processing tasks.

The anticipated outcome of the SFGS transcends mere technological innovation—it promises nothing short of a revolutionary evolution in fruit grading systems. A highly efficient, accurate, and versatile system, the SFGS stands poised to eradicate the need for manual intervention in fruit grading. Through the seamless fusion of automation and AI tools, the system is destined to elevate efficiency, precision, and adaptability within the agricultural supply chain. Rooted in carefully crafted engineering decisions, encompassing tailored hardware specifications and layered software architecture, the SFGS emerges as a beacon of groundbreaking advancement in agricultural technology. With the potential to significantly enhance the efficiency of fruit grading processes, this system holds the promise of delivering widespread benefits to stakeholders across the agricultural industry, heralding a new era of technological prowess in agriculture.

Within the Smart Fruit Grading System, a diverse array of hardware components plays integral roles. The system incorporates a range of motor types, employing Servo motors for gate control and DC motors to facilitate the movement of the fruit conveyor. Notably, a collection of cameras, including an X-ray camera, is employed to capture crucial imaging data. At the core of the system's operations is a microcontroller, serving as the primary intelligence hub that orchestrates and coordinates the functionalities of the various components.

2. Document Overview

The Smart Fruit Grading System (SFGS), a pinnacle of agricultural innovation, is engineered to autonomously revolutionize fruit grading processes. Functioning as a crucial link between farms and distributors, the system operates within specialized operational rooms dedicated to fruit cleaning and boxing. Targeting the manual fruit grading stage, it caters to small farms, distributors, and global players alike, offering a versatile solution that enhances the efficiency of the agricultural supply chain. In a user scenario, the system processes fruits sequentially through input gates, employs advanced cameras, including an X-ray camera, and utilizes artificial intelligence to grade fruits into four categories: high, medium, low, and inedible. The output gates then direct fruits based on their grades, showcasing a streamlined and automated fruit grading process.

The hardware architecture is robust, with a microcontroller at its core managing cameras, gates, and conveyors. Cameras capture images of fruits, sensors trigger photography, and an X-ray camera provides a comprehensive inspection. Servo motors control gates, while a dedicated power supply unit ensures the system's smooth functionality. This interconnected hardware architecture is visualized in Figure 3, portraying the system's operational dynamics.

The hardware components, detailed in Figure 3, encompass a central microcontroller, cameras with varied functionalities, gates, sensors, and a conveyor system adhering to industry standards. Each component plays a vital role, ensuring the systematic movement and accurate grading of fruits.

The software architecture is depicted in a flowchart (Figure 7), offering a high-level perspective of the system's processes and interactions. Key components include the UI Component, Reset Component, AI Model Component (Fine-Tuned Densent-201, and Motion Detection Algorithm), and Camera Interface Component. The Ubuntu operating system, version 22.04 LTS, serves as the software foundation, ensuring stability and compatibility for optimal performance. Our technical choices are deliberate, with DenseNet-201 chosen for the AI model due to its accuracy and efficiency in fruit grading. The layered software structure allows for easy upgrades, and hardware specifications are tailored to meet the demands of the AI model and overall system functionality.

Assumptions revolve around a stable operating environment with consistent lighting, reliable sensor and camera data, and an efficient conveyor system. These assumptions guide our engineering decisions, reflecting a systematic and innovative approach to the Smart Fruit Grading System.

The overview of our system is illustrated in the following figure. Fruits are sequentially fed into the system via a moving conveyor. As a fruit moves beneath the cameras, a motion detection algorithm activates the cameras to capture seven images. These images are then sent to the AI model for analysis. The model examines the images and produces a class label out of four possible options. Subsequently, the obtained label is utilized to update the current statistics and direct the fruit to the assigned line.

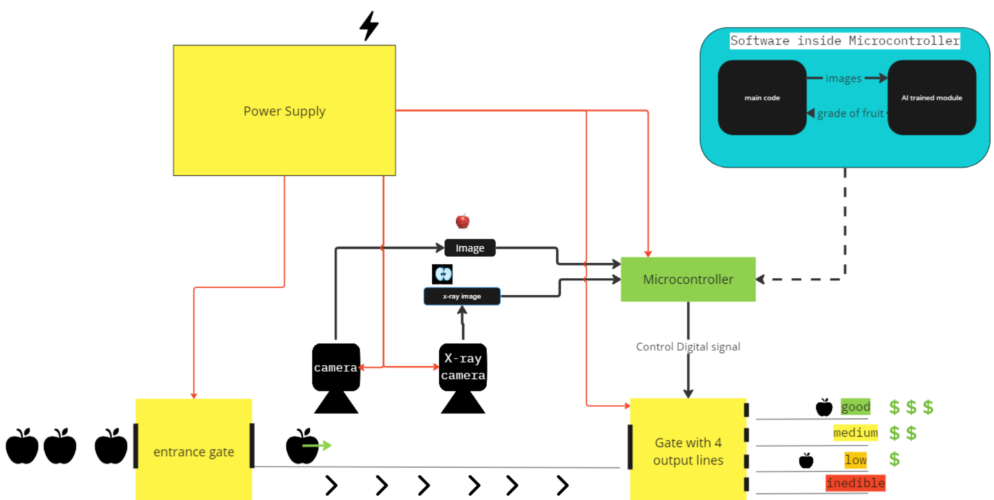

Figure 1. Architecture Overview of the SFGS.

3. Abbreviations and Glossary

#	Acronym	Full Form
A1.	AI	Artificial Intelligence
A2.	DenseNet	Densely Connected Convolutional Networks
A3.	DFM	Data Flow Models
A4.	DL	Deep Learning
A5.	EDD	Engineering Design Document
A6.	IrDA	Infrared Data Association
A7.	PMP	Project Management Plan
A8.	PNG	Portable Network Graphics
A9.	PWM	Pulse Width Modulation
A10.	SFGS	Smart Fruit Grading System
A11.	SPI	Serial Peripheral Interface

A12.	TIFF	Tagged Image File Format
A13.	UML	Unified Modeling Language
A14.	UC	User Case
A15.	USB	Universal Serial Bus

4. References

#	Document Identifier	Document Title
R1.	T231-COE384-PMP-002	Project Management Plan
R2.	T231-COE384-BC-002	Business Case

5. Standard and regulatory References

#	Identifier	Link to the standard/regulatory
STD1.	USB 3.0	USB 3.0 Standard and Regulation Document
STD2.	SPI	SPI Standard and Regulation Document
STD3.	IrDA	IrDA Standard and Regulation Document
STD4.	PNG	PNG Standard and Regulation Document
STD5.	UML	UML Standard and Regulation Document
STD6.	TIFF	TIFF Standard and Regulation Document
STD7.	OCED	International Standards for Fruit and Vegetables | OECD iLibrary (oecd-ilibrary.org)

6. Architecture

6.1. Architecture Overview

The primary purpose of the smart fruit grading system is to autonomously assess the quality of fruit, eliminating the need for manual human intervention. This advanced system integrates artificial intelligence (AI) tools alongside various industrial components to revolutionize and automate processes within the agricultural industry. Positioned within the food supply chain, it operates seamlessly between the farm and fruit distributors, serving as a versatile solution for both parties. The system is specifically designed for application in operational rooms dedicated to cleaning and boxing fruits.

The system's targeted area of intervention is the manual fruit grading stage, situated between the cleaning and boxing processes. Small farms engaged in self-distribution, as well as distributors handling ungraded fruits for subsequent cleaning, grading, and boxing, can benefit from this innovative technology. Notably, global distributors stand to gain the most from the system's capabilities.

In essence, the system operates by receiving fruits in a serial and sequential manner. Following this input, it categorizes the fruits into four levels: high, medium, low, and inedible. Each grade is then directed to a specific gate corresponding to its quality. Figure 1 illustrates the system's fundamental concept, depicting the input of fruits and the output through various gates based on their respective grades. This comprehensive architecture overview underscores the system's potential to streamline and enhance fruit grading processes, contributing significantly to the efficiency of the agricultural supply chain.

In the Smart Fruit Grading System, a combination of cameras, including a set of standard cameras and a single X-ray camera, is integral to the process. Specifically, the inclusion of an X-ray camera is crucial for comprehensive fruit inspection. These cameras collectively capture images of the moving fruits positioned above a conveyor belt, transmitting the visual data to a central microcontroller.

The microcontroller, equipped with an AI-trained module, plays a pivotal role in determining the grade of each fruit. Following the evaluation, the microcontroller takes charge of the output gate, directing the fruit to its designated line based on its quality. As all fruits traverse above the conveyor belt, the continuous forward movement is orchestrated by a conveyor roller powered by an electric motor.

Dynamic control of the gates is achieved through servo motors, ensuring precise adjustments in the fruit flow. To sustain the entire system's functionality, a dedicated power supply unit distributes power to all electrical components, spanning from the microcontrollers to the various motors. A graphical representation of this interconnected process is available in Figure 3, providing a visual guide to the Smart Fruit Grading System's operational dynamics.

Within the hardware architecture, diverse components play essential roles, with the microcontroller standing out as the primary and central element. Functioning as the controller for all motors and cameras, the microcontroller utilizes USB 3.0 to seamlessly receive images from the camera-standard-X's, which capture in the PNG format. Notably, an X-ray camera, named camera-Xray-1, employs the SPI protocol to transmit images in the TIFF format. Six camera-standard-X's strategically cover fruits from various angles along the Positive-x-axis, negative-x-axis, Positive-y-axis, and more, as elucidated in Figure 4. The challenging negative-z-axis position, located beneath the fruits, is addressed by camera-standard-6, strategically placed below the transparent plastic conveyor belt, known as "conveyor-belt-plastic."

The system features two gates, an input gate labeled "input-gate," and an output gate labeled "output-gate." While both share similar structures, the output-gate is slightly advanced. The input-gate incorporates a servo motor with a driver, facilitating the opening of the entrance gate for new fruits. In contrast, the output-gate directs fruits to specific lines based on their grade. In different modes, the gates are effectively closed. The receiver for the DC-Motor-Driver uses PWM to communicate with a 12 Volt power source.

Sensors play a crucial role by determining whether fruits are positioned for image capture, acting as triggers for the photography process.

The conveyor system adheres to industry standards, with the objective of seamlessly transporting fruits from the system's start to its end. Comprising a DC motor labeled "DC-Motor" controlling a "Conveyor-roller," which enhances the movement of the transparent plastic conveyor belt labeled "conveyor-belt-plastic." The unique transparency of the conveyor-belt-plastic allows for image capture below the fruits, with all motors employing the PWM protocol for efficient communication.

Figure 3 provides a comprehensive illustration of the system composition.

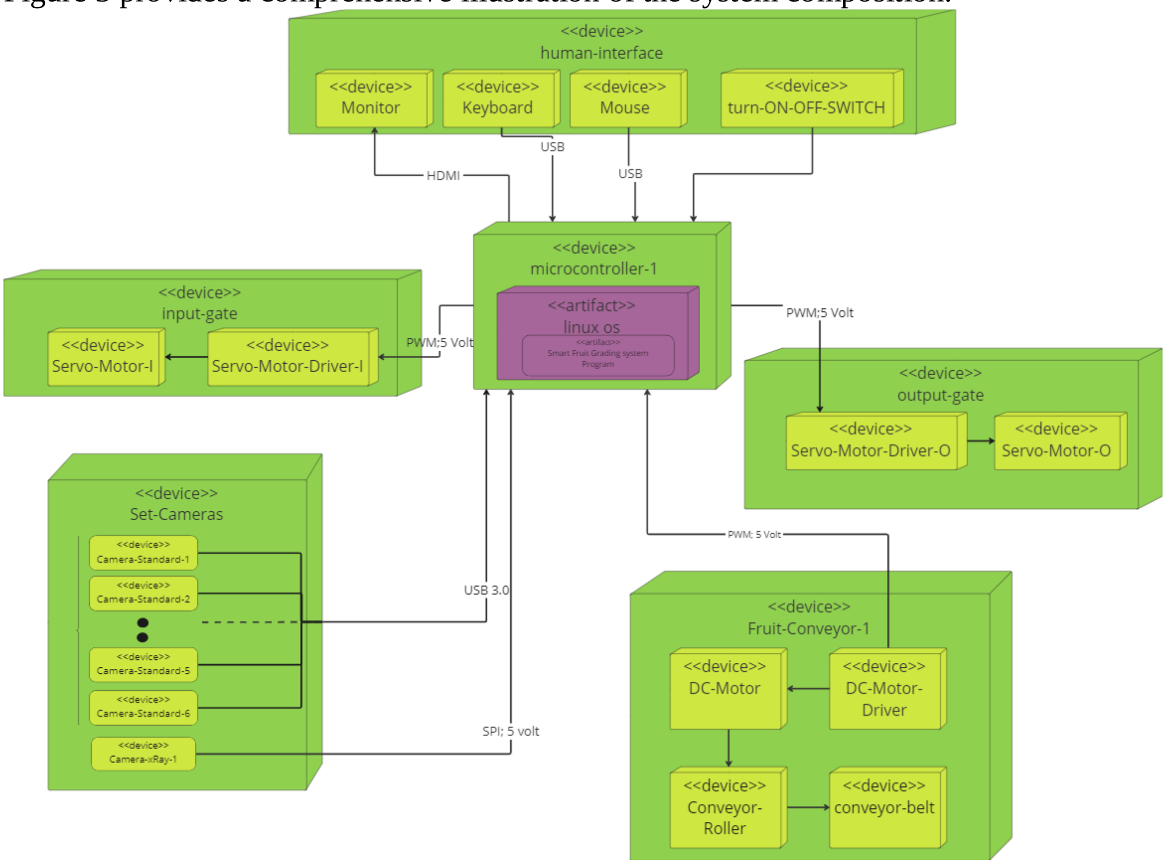

Figure 3. Deployment Diagram for SFGS Hardware

The operational process of the system unfolds systematically. Initially, the input gate opens, facilitating the transfer of fruits to the fruit-conveyor. Subsequently, the fruit-conveyor propels the fruits forward, guiding them to the set-camera-standard. Upon reaching this point, cameras capture images from all six angles, and these images are then transmitted to the microcontroller. The next stage involves movement to the x-ray-room, where x-ray photos are taken and relayed to the microcontroller for further analysis. Leveraging both the images and an AI-trained module, the microcontroller accurately determines the grade of the fruit. The subsequent step involves the fruit-conveyor transporting the graded fruit to the output gate. The microcontroller takes charge of the output-gate, directing the fruit to the designated line. For a visual elucidation of this process, refer to Figure 4, which provides a comprehensive and illustrative representation.

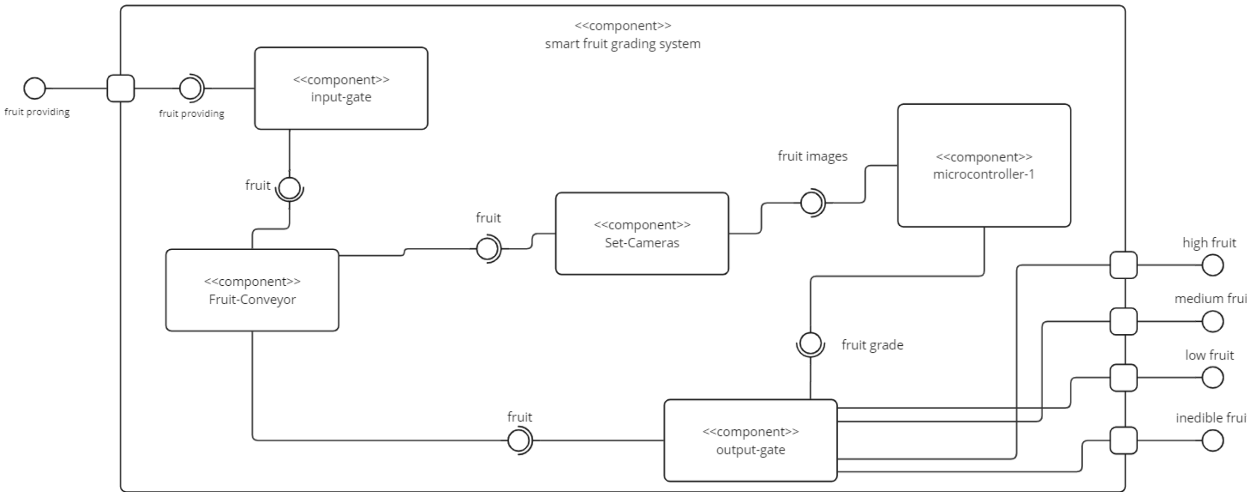

Figure 4. Component Diagram for SFGS

6.1.1. Microcontroller

The microcontroller has an integration ID microcontroller-1. It holds a unique position as the sole controller within the system, designed to oversee and manage the entire system while implementing sophisticated computer vision capabilities. This pivotal component houses all software elements, ranging from the AI module to various interfaces. The microcontroller's technical specifications boast a Broadcom BCM2712 2.4GHz quad-core 64-bit Arm Cortex-A76 CPU, equipped with 512KB per-core L2 caches and a 2MB shared L3 cache. This robust CPU adeptly handles the execution of the AI-trained module and the overall system control. The system's 8GB RAM efficiently caters to both the operating system and the main code requirements. Utilizing a 128GB SD card as the primary disk streamlines software loading and updates, requiring a dedicated SD card reader interface.

The microcontroller features an array of hardware interfaces, including 6 USB 6.0 ports for the set-camera-standard, as well as two additional USB 2.0 ports for interfacing with a keyboard and mouse to interact with the operating system. An HDMI port facilitates display connections. Furthermore, specific SPI pins are allocated for the x-ray camera, while two reserved pins for PWM in 5 Volts cater to both the input-gate and output-gate. An additional PWM pin operates at 12 Volts. Additionally, the microcontroller is equipped with 10 pins dedicated to two IrDAs, each utilizing five pins for seamless communication.

Also, it has an additional Pin for turning ON/OFF the system. The pin is PULLED_UP. The idea is that the Pin is linked with a turn-ON-OFF-Switch that if Switch up will link the Pin to the ground. if the Pin is in the ground state, the system is OFF. if the Pin is in the High voltage state, the system will be ON.

6.1.2. Turn-ON-OFF-switch

It's a toggle switch for the system to be Turn ON/OFF switch. It also has the hardware Debounce to save the electric circuit. Its resistance to a range of -40C to 100C heat.

6.1.3. Driver

6.1.3.1. Servo Motor Driver

The Servo Motor Driver, which its integration ID is Servo-Motor-Driver-I for the input gate one and Servo-Motor-Driver-O for the output gate. The two drivers have the same specifications. The driver is a L298N model, an integral component in the control and operation of DC motors within the Smart Fruit Grading System. Its primary purpose lies in acting as a bridge between the microcontroller and the DC motor, facilitating the regulation of speed and direction.

In terms of software component interaction, the DC Motor Driver receives signals from the microcontroller, interpreting them to generate precise instructions for the connected DC motor. This translation mechanism ensures efficient coordination between the microcontroller and the motor, enabling seamless control.

Exploring its technical characteristics, the L298N motor driver supports a wide voltage range from 5V to 35V, providing flexibility in power supply. With a current handling capacity of up to 2A per channel, the driver ensures efficient power delivery to the connected motor.
Operating at a logical voltage of 5V and a logical current ranging from 0 to 36mA, the driver is equipped with two channels, forming a dual H-bridge configuration. This design allows independent control of two motors, enhancing versatility in motor control applications.

The DC Motor Driver accommodates high-level control signals in the range of 2.3V to Vss and low-level control signals from GND to 1.5V. This broad input range ensures reliable signal processing, contributing to the precision of motor control. Operating within a temperature range of 0°C to 80°C, the driver exhibits adaptability to diverse environmental conditions, enhancing its reliability in varied operating environments.

In terms of network hardware interfaces, the DC Motor Driver seamlessly connects to the microcontroller using General Purpose Input/output (GPIO) pins. This straightforward interface establishes a reliable communication link for control signals, enhancing integration with various microcontroller platforms.

The software component connected to is GateInterface.

6.1.3.2. DC motor Driver

The DC Motor Driver with an integration ID DC-Motor-Driver operates as a PWM-driven controller, receiving 5V and transmitting PWM: 24V to regulate the DC motor responsible for controlling the fruit-conveyor. This driver establishes a direct link between the microcontroller and the conveyor motor, facilitating precise control over the conveyor system. The software component connected to is GateInterface.

6.1.4. Motors

6.1.4.1. Servo Motor

The Smart Fruit Grading System relies on a critical component known as the Servo Motor, specifically identified by Integration ID: Servo-Motor-I for the input gate and Servo-Motor-O for the output gate. These two servo motors are identical in their specifications. These motors play a pivotal role in orchestrating the systematic movement of fruits along the processing line within the system.

The motors operate under the guidance of the central processing unit (CPU) of the Smart Fruit Grading System. They receive instructions from the system's software, which encompasses algorithms for fruit classification, sorting decisions, and adjustments to the conveyor speed. This intricate connection between software and hardware is vital for the precise control and movement of fruits throughout the grading process.

In terms of technical characteristics, the Servo Motor adopted for this purpose is specifically tailored for precision control and speed adjustments. Its design aligns with the dynamic requirements of fruit movement, ensuring accurate positioning and responsiveness. The motor operates on 24V DC, 5A, a carefully chosen voltage and current specification that balances power efficiency with the torque demands of the conveyor system. This balance provides sufficient power for the smooth and efficient operation of the conveyor.

Safety features are integrated into the design, including overcurrent protection and an Emergency Stop (E-stop) capability. These features are crucial for preventing motor damage during overcurrent conditions and enabling the swift halting of conveyor operation in emergencies, prioritizing both equipment integrity and overall safety.

Considering environmental factors, the Servo Motor is designed to operate within an operating temperature range of 0°C to 40°C. This design choice ensures reliable performance even in the diverse environmental conditions commonly found in fruit processing facilities.

In terms of network hardware interfaces, the motor adopts the Modbus protocol for communication with the central controller. Modbus is chosen for its simplicity, reliability, and widespread adoption in industrial applications. The motor utilizes a wired connection to the central processing unit (CPU), ensuring a stable and robust link for communication. Furthermore, it features an Ethernet interface, leveraging the standardization and compatibility of Ethernet connectivity with modern industrial communication systems. Operating at a speed of 100 Mbps, the network ensures swift and responsive communication between the conveyor motor and the central controller, contributing to the overall efficiency of the Smart Fruit Grading System.

6.1.4.2. DC Motor

The Brushless DC Motor, identified as DC-Motor plays a critical role in the Smart Fruit Grading System by serving as the primary driving force for the conveyor belt, ensuring the smooth and efficient movement of fruits throughout the grading process. Its integration with the Motor Driver is key to this functionality, as it enables the motor to receive precise commands that dictate its speed and direction. This seamless interaction between the Brushless DC Motor and the Motor Driver enhances the overall control and precision of the system, contributing to its effectiveness in fruit grading.

The technical specifications of the Brushless DC Motor include the following operating at a reference voltage of 24V, the motor demonstrating a current of 3A, with a peak current reaching 6A. Additional characteristics include a no-load current of 1A, a resistance of 2 ohms, inductance of 5 mH, speed capabilities of up to 3000 rpm, and a temperature tolerance ranging from -10°C to 50°C. Crucially, the motor delivers a continuous torque of 2 Nm, driven by a motor constant of 24 V/A. These specifications collectively highlight the motor's adaptability and suitability for its pivotal role in the Smart Fruit Grading System, ensuring it meets the dynamic demands of fruit processing with precision.

In contrast to the Servo Motor, the Brushless DC Motor operates independently, devoid of network hardware interfaces. This intentional design choice underscores the motor's reliability in its dedicated function of driving the conveyor belt. By avoiding network dependencies, the DC Motor contributes to the overall efficiency and robustness of the fruit grading system, simplifying its operation, and enhancing its dependability in various processing conditions.

6.1.5. Cameras

The set of all cameras is identified as Set-Cameras and the software component connected to all cameras in the system is CameraInterface

6.1.5.1. Set-camera-standard

The standard camera, identified by the Integration ID Camera-Standard-X, is an integral component of the Smart Fruit Grading System. Its primary function is to capture high-resolution visible light images of fruits as they move along the conveyor belt. This documentation provides a comprehensive overview of the technical specifications, performance features, environmental considerations, integration details, and additional functionalities of this sophisticated imaging component.

The Smart Fruit Grading System incorporates a total of six cameras, each strategically positioned to operate from specific angles along the conveyor belt. These cameras, collectively identified as Camera-Standard-X, where X starts from 1 to 6, work in tandem to capture comprehensive images of the fruits in transit. The distribution arrangement ensures thorough coverage, allowing for a holistic examination of each fruit from various perspectives. This multi-camera setup enhances the system's ability to capture diverse features, contributing to more accurate and detailed assessments during the grading process. The specific identification, Camera-Standard-X, facilitates streamlined monitoring and management of each camera's role and performance within the overall imaging framework.

In terms of technical specifications, the camera boasts 240P resolution, selected to capture intricate details of fruit surfaces for precise grading and defect detection. CMOS sensor technology, with a sensor size of 1/1.7 inches, balances image quality and power efficiency. The choice of a fixed focal length lens enhances consistency and reliability, eliminating the need for constant adjustments during operation.

Performance specifications include a 60fps frame rate, enabling the camera to capture a continuous stream of images for efficient fruit processing. Automatic focusing, autoexposure, and optimized light sensitivity contribute to the camera's versatility across various produce types and lighting conditions. The camera interfaces with a USB 3.0 for high-speed data transfer to the Microcontroller, ensuring real-time processing and analysis with a 5 Gbps data transfer rate.

Power supply specifications include a 12V DC .5 Amp Output, 100-240 VAC 50/60Hz Input, and a 2.1mm plug. The camera operates in an environmental temperature range of -10°C to 50°C, with a storage range of -20°C to 70°C.

Regarding integration and compatibility, the camera seamlessly integrates with microcontrollers, supporting specified software or image processing libraries for efficient data processing and analysis. User interface compatibility allows operators to monitor and adjust the grading process, enhancing user experience and system control.

Durability and reliability are ensured through the camera's robust construction, capable of withstanding the demanding conditions of fruit processing facilities. Regular calibration intervals of every 6 months to a year maintain optimal performance over time.

In terms of image acquisition and processing, the standard camera captures fruit images transmitted to the Image Acquisition Module. This software processes images using various techniques, including color analysis, texture analysis, and shape recognition, extracting information such as size, color, texture, and overall condition.

The extracted features from the standard camera significantly contribute to the fruit grading process. The fruit's size and shape determine its category, while color and texture analysis identify defects like blemishes, bruises, or ripeness inconsistencies. The standard camera plays a crucial role in defect detection, ensuring that only edible-quality fruits proceed for further processing or packaging.

6.1.5.2. X-ray Camera

The X-ray camera, identified by the Integration ID Camera-xRay-1, is a pivotal component within the Smart Fruit Grading System, specializing in non-destructive internal inspections of fruits. This documentation provides an in-depth exploration of its engineering specifications, covering resolution, sensor technology, energy efficiency, and integration nuances, establishing a solid foundation for seamless integration with microcontrollers.

In terms of engineering specifications, the X-ray camera boasts fine-tuned resolution for high-resolution X-ray imaging, capturing intricate internal details crucial for defect detection at 320x240 pixels. It incorporates an advanced CMOS X-ray sensor for efficient penetration, utilizing a variable sensor size to accommodate diverse fruit dimensions and a transparent lens to minimize interference, ensuring unaltered X-ray beam transmission for precise internal imaging.

Performance specifications include a variable frame rate ranging from 5fps to 30fps, adaptable to optimize precision in internal imaging and accommodate real-time processing needs. Automatic depth adjustment enhances clarity, catering to variations in fruit size and density, while adjustable energy settings facilitate optimal X-ray penetration for different fruit types and densities.

Connectivity features include the use of the SPI protocol for seamless communication with microcontrollers, achieving a data transmission rate of 5 Gbps for rapid transfer of X-ray images. The X-ray camera operates on a 3.3V power supply with a current requirement of 500mA, aligning with the microcontroller's power specifications. Specific control signals, including trigger and configuration signals, enable precise microcontroller control over image capture and camera settings.

In terms of microcontroller compatibility, the camera is engineered to work seamlessly with widely used microcontrollers, such as the Raspberry Pi 4 Model B, featuring an SPI interface for straightforward integration. It provides eight GPIO pins, allowing the microcontroller to control external features or receive status signals, enhancing flexibility in system configuration. The camera adheres to the Python SPI library, facilitating easy integration with the microcontroller's programming environment.

Power requirements include an input power of 180-264 Vac, single-phase, and an output voltage range of 0-160 kV with a corresponding current range. The camera incorporates voltage regulation features for both line and load changes, ensuring stability during operation. The onboard ARM Cortex-M4 processor enables preliminary data processing, including image filtering and compression before transmission to the microcontroller, transmitting X-ray images in a compressed JPEG format for optimized data transfer efficiency and compatibility with the microcontroller's image processing capabilities.

User interaction features include status feedback signals through dedicated GPIO pins, indicating operational status, errors, and diagnostic information to the microcontroller. The camera also receives user commands via SPI communication, allowing dynamic adjustments to imaging settings, capturing triggers, and other operational commands.

In consideration of environmental factors, the X-ray camera is designed to operate within an extended temperature range of -10°C to 60°C, ensuring reliable performance in varying environmental conditions. It is engineered to withstand vibrations up to 5 g-force and shocks up to 50g, ensuring robustness and stability during operation in industrial environments.

6.1.6. Physical structure

Overview:
The Physical Structure of the Smart Fruit Grading System is comprised of the Conveyor Belt Frame, Conveyor Belt Rollers, and Conveyor Belt, meticulously engineered with specific numerical values and detailed specifications to ensure stability, endurance, and efficient fruit conveyance.

6.1.6.1. Conveyor Belt Frame

The Conveyor Belt Frame, identified as Conveyer-Belt-Frame is an essential component in the Smart Fruit Grading System, serves a foundational purpose by providing stability during dynamic fruit grading operations. Constructed from high-tensile steel, this robust structural support framework is engineered to bear a load capacity of 5,000 kg, ensuring it meets specified requirements for stability.

Primarily a mechanical element, the Conveyor Belt Frame interacts with software components only in specific scenarios involving advanced automation. The conveyor Belt Frame typically lacks specific network interfaces in standard operations.

6.1.6.2. Conveyor Belt Rollers

The Conveyor Belt Rollers, identified as Conveyer-Belt-Rollers are within the Smart Fruit Grading System, also they are purposefully engineered components designed to facilitate the seamless movement of fruits. Their precision-crafted rotating rollers, crafted from hardened steel, contribute to operational efficiency and prolonged durability. With specified diameters ranging from 10 cm to 15 cm, these rollers are tailored to operational needs, ensuring optimal fruit conveyance during the grading process. Like the Conveyor Belt Frame, the Conveyor Belt Rollers generally interact with software components in scenarios involving advanced automation features. Despite their crucial role in the system, these rollers typically do not feature direct network interfaces, aligning with a focus on simplicity and reliability in their contribution to the overall efficiency of fruit conveyance within the Smart Fruit Grading System.

6.1.6.3. Conveyor Belt

The Conveyor Belt, identified as Conveyer-Belt designated as a critical component within the Smart Fruit Grading System, plays a pivotal role in ensuring the seamless transport of fruits, emphasizing durability and efficient movement. Comprising a continuous transport belt, it is crafted from high-strength, wear-resistant materials, including a polyester and rubber composite. Configured with a width of 30 cm and a variable speed ranging from 0.5 m/s to 1.5 m/s, the Conveyor Belt is tailored to meet specific grading line requirements. In terms of software component interaction, the Conveyor Belt receives control signals from the main grading system software, facilitating precise functions such as start, stop, and speed adjustments. Also, it is required that the conveyor belt is made of a transparent material to be able to capture the beneath perspective of the fruit.

6.1.7. Power supply

This power supply is the powerhouse for the entire system, ensuring smooth operation. It starts by energizing two ports dedicated to servo motor drivers, each operating at 24V and 5A. Additionally, there are ports tailored for 24V and 3A to drive DC motor drivers. The power supply also features six ports supplying 12V and 5A for cameras, and a separate 12V and 3A port for the microcontroller. Simply plug the power supply into a standard outlet with 220V, 60Hz, and a capacity of 10A to get everything up and running. With this setup, each component receives the right amount of power for optimal functionality.

6.2.Software architecture overview

The following flowchart presents a high-level perspective of the Smart Fruit Grading System, offering an overview of its fundamental processes and interactions. From the initial stage of image capture to the outcomes of grading predictions and reset operations, this visual representation encapsulates the core functionalities and relationships within the system. Providing a broad understanding of the system's workflow, the flowchart serves as a concise yet comprehensive guide to the various components and their collaborative roles in ensuring the seamless operation of the fruit grading system.

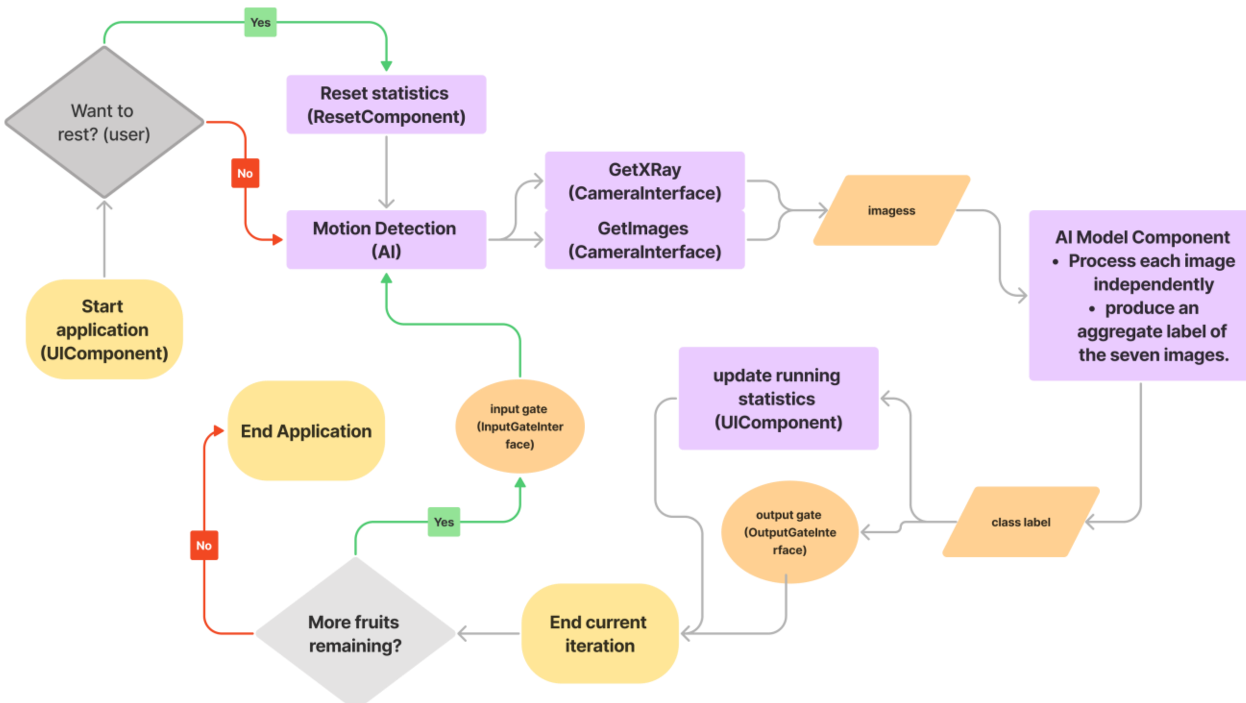

Figure 5. Flowchart Depicting the Overall Design and Interactions Among the Smart Fruit Grading System Components

Interactions:

- The UI Component communicates with the AI Model Component and triggers reset operations through the Reset Component.
- The Reset Component listens for reset commands from the UI Component.
- The AI Model Component receives input data from the Camera Interface Component and provides grading predictions.
- The Camera Interface Component sends image files to the AI Model Component.

Operating Systems:

The software is intended to operate on the Ubuntu operating system, specifically version 22.04 LTS. Ubuntu, a popular and user-friendly Linux distribution, serves as the foundation for hosting and executing the various components of the fruit grading application. Leveraging the stability and reliability of Ubuntu 22.04 LTS ensures compatibility and seamless functionality, creating an optimal environment for the app's performance.

6.2.1. User Interface Component

- **Identification:** UIComponent
- **Purpose:**
    - Provides the graphical user interface for the desktop application.
    - Displays the total number of boxes for each grade and a reset button, visual representation.
    - Allows user interaction for initiating reset operations.
- **Interfaces:**
    - Communicates with the Grading and Sorting Component to retrieve grading information.
    - Triggers reset operations by interacting with the Reset Component.
- **Network Interfaces:**
    - No external network communication for the UI component.
- **Hardware Resources:**
    - Average RAM Usage: 100 MB
    - Peak RAM Usage: 200 MB
    - Average CPU Usage: Low to moderate (5% - 10%)
    - Peak CPU Usage: Moderate (11% - 15%)
    - Average CPU Frequency: 2.5 GHz
    - Peak CPU Frequency: 3.5 GHz
    - Average Duration of Peak CPU Usage: 10 seconds
    - Disk Space for Permanent Data: Negligible (5 MB for settings and configuration)
    - Disk Space for Cache Data: Negligible (10 MB for UI-related cache)

6.2.2. Reset Component

- **Identification:** ResetComponent
- **Purpose:**
    - Manages the reset functionality for the grading statistics.
- **Interfaces:**
    - Listens for reset commands from the UI Component.
- **Network Interfaces:**
    - No external network communication.
- **Hardware Resources:**
    - Average RAM Usage: Low (~50 MB)
    - Peak RAM Usage: Low during reset operations (~90 MB)
    - Average CPU Usage: Low (2% - 5%)
    - Peak CPU Usage: Low during reset operations (8% - 10%)
    - Disk Space for Permanent Data: Negligible (10 MB)
    - Average CPU Frequency: 2.0 GHz

- Peak CPU Frequency: 3.0 GHz
- Average Duration of Peak CPU Usage: 2 seconds
- Disk Space for Cache Data: Negligible (5 MB)

6.2.3. AI Model Component

The AI model component harnessing the power of deep learning techniques, specifically DenseNet-201, serves as the masterpiece of the software aspect of the project. The PyTorch package will be utilized to implement the functionalities of the component. The following table shows all the configurations associated with the deep learning model.

Table 1. Parameters Settings of the Deep Learning Model

Setting	Value
Architecture Name	DenseNet-201
Exptected Image size	(224, 224)
Total Layers	201
Dense Blocks	4
Layers per Dense Block	Varies
Growth Rate	Varies
Bottleneck Compression Factor	0.5
Transition Layer Compression	1x1 Convolution + 2x2 AvgPooling
Activation Function	ReLU
Dropout Rate	0.3
Weight Initialization	He initialization
Batch Normalization	Applied after Convolutional operations

- **Identification:** AIModelComponent
- **Purpose:**
  - Runs the deep learning model (DenseNet201) for grading fruits.
  - Receives input data from the Camera Interface Component.
  - Provides grading predictions.
- **Interfaces:**
  - Communicates with the Camera Interface Component for receiving input data.
  - Provides grading predictions to the Gate Interface Component.
- **Network Interfaces:**
  - Communicate with the Gate Interface Component.
- **Hardware Resources:**
  - Average RAM Usage: High (1 GB)
  - Peak RAM Usage: High during model inference (2 GB)
  - Average CPU Usage: High (20% - 30%)
  - Peak CPU Usage: Very high during model inference (40% - 50%)
  - Average CPU Frequency: 3.5 GHz
  - Peak CPU Frequency: 4.5 GHz
  - Average Duration of Peak CPU Usage: 1 second

- Disk Space for Permanent Data: Python, NumPy, PyTorch, Model weights, configurations, OpenCV, 1.3 GB
- Disk Space for Cache Data: Minimal (caching recent predictions, 150 MB)

6.2.4. Camera Interface

- **Identification:** CameraInterface

The CameraInterface, a lightweight component, is utilized for seamless communication with both camera-standard-X's and camera-Xray. It efficiently retrieves an array of images from the cameras and uploads them to the AI-trained module for computer vision analysis, ensuring minimal resource consumption.

- **Hardware Resources:**
  - Average RAM Usage: Low(250 MB)
  - Peak RAM Usage: High during model inference (260 MB)
  - Average CPU Usage: High (10%)
  - Peak CPU Usage: Very high during model inference (20%)
  - Average CPU Frequency: 100 MHz
  - Peak CPU Frequency: 200 MHz
  - Average Duration of Peak CPU Usage: 2 second
  - Disk Space for Cache Data: Minimal (caching recent predictions, 150 MB)

6.2.5. Gate Interface

- **Identification:** GateInterface

The idea of the GateInterface is to manage both input-gate and output-gate in one class. Generally, it will consume low Hardware resources. since its a low-level interface, just utilizing PWM modulation to control the gates.

6.2.6. Fruit conveyor Interface

- **Identification:** FruitConveyorInterface

The case of fruitConveyorInterface, it will utilize PWM; 12 Volt pin to control the DC-Motor-Driver, hence controlling the conveyor. either forward or stopping. the hardware Resources is negligible.

6.3.Third Party Software

6.3.1. Ubuntu (Operating System)

- **Identification and Version:** Ubuntu, version 22.04 LTS
- **Purpose for Your Project:**

  - Ubuntu serves as the operating system for hosting and running the desktop application and AI model components of the fruit grading and sorting system.
- **Source:**
  - Ubuntu is an open-source Linux distribution developed and maintained by Canonical Ltd.
- **Maintenance:**
  - Ubuntu is actively maintained by Canonical Ltd. with regular Long-Term Support (LTS) releases.
- **Resources:**
  - Moderate RAM usage for desktop environment and application processes (4 GB).
  - Minimal CPU usage for system processes.
  - Disk space for storing the operating system and application files (25 GB).
- **Interfaces and Data Flows:**
  - Interfaces with the desktop application and AI model components for providing a runtime environment.
  - Data flows involve communication between the operating system and the running application components.

6.3.2. Python (Programming Language):

- **Identification and Version:** Python, version 3.8.15
- **Purpose:**
  - Python serves as the programming language for developing various components of your fruit grading and sorting system, including the desktop application and AI model.
- **Source:**
  - Python is an open-source programming language developed and maintained by the Python Software Foundation and a large community of contributors.
- **Maintenance:**
  - Python is actively maintained with regular releases and updates by the Python Software Foundation.
- **Resources:**
  - Minimal RAM usage for basic script execution.
  - Minimal CPU usage for running Python scripts.
  - Disk space for storing Python interpreter and scripts (200 MB).
- **Interfaces and Data Flows:**
  - Interfaces with the development environment for writing and executing code.
  - Data flows involve communication between Python scripts and other components of the fruit grading and sorting system

6.3.3. PyTorch (Deep Learning Framework):

- **Identification and Version:** PyTorch, version 2.0.1.

- **Purpose:**
  - PyTorch will be used to implement and deploy the DenseNet-201 deep learning model for classifying the images into four classes.
- **Source:**
  - Developed by Facebook's AI Research lab (FAIR).
- **Maintenance:**
  - PyTorch is actively maintained by the PyTorch community and Facebook.
- **Resources:**
  - High CPU (20% - 30%) usage during model inference.
  - Disk space for storing the PyTorch library and model checkpoints (800 MB).
- **Interfaces and Data Flows:**
  - Interfaces with the AI Model Component for building, and inference of the DenseNet-201 model.
  - Data flows involve input images for inference, specifically tailored for fruit grading (Seven instances per fruit, six normal images, and one X-ray image).

6.3.4. NumPy (PyTorch Dependency)

- **Identification and Version:** NumPy, version 1.24.3
- **Purpose (as a PyTorch Dependency):**
  - NumPy is a fundamental dependency for PyTorch, providing support for multi-dimensional arrays and mathematical functions. PyTorch uses NumPy arrays as a bridge for exchanging data with its tensor data structures.
- **Source:**
  - NumPy is an open-source project, developed and maintained by the NumPy community.
- **Maintenance:**
  - NumPy is actively maintained by the open-source community with regular releases.
- **Resources:**
  - Minimal RAM usage for basic numerical operations (1 GB).
  - Minimal CPU usage for standard array manipulations.
  - Disk space for storing the NumPy library (100 MB).
- **Interfaces and Data Flows:**
  - There is no direct interaction with the package. It is being installed because PyTorch relies on it.

6.3.5. OpenCV (Open-Source Computer Vision Library):

- **Identification and Version:** OpenCV, version 4.8.0
- **Purpose**
  - OpenCV will be used for image processing tasks, such as pre-processing and feature extraction, as part of the fruit grading and sorting system.

- **Source:**
  - OpenCV is an open-source project, developed by the OpenCV community.
- **Maintenance:**
  - OpenCV is actively maintained by the open-source community.
- **Resources:**
  - Moderate RAM usage for image processing tasks.
  - Minimal CPU usage during basic image processing tasks.
  - Disk space for storing the OpenCV library (200 MB)
- **Interfaces and Data Flows:**
  - Interfaces with image data for processing in the Grading and Sorting Component.
  - Data flows involve input images for pre-processing and feature extraction.

7. Requirement Management

7.1. Use Cases

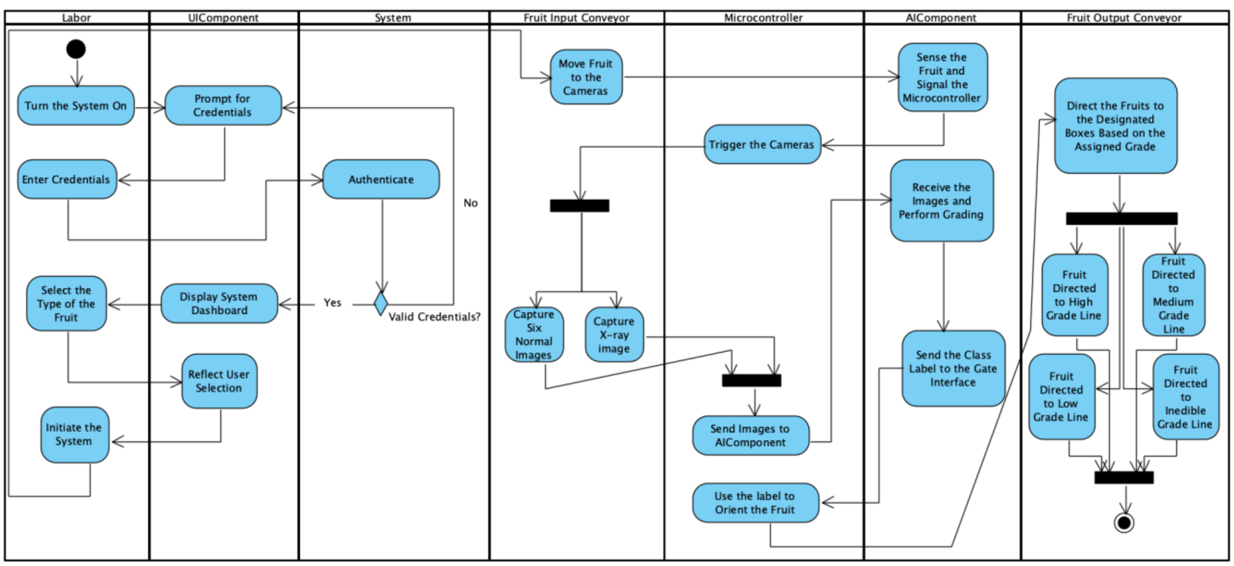

Figure 6. Overall Activity Diagram Depicting the Interactions Among the System's Various Components.

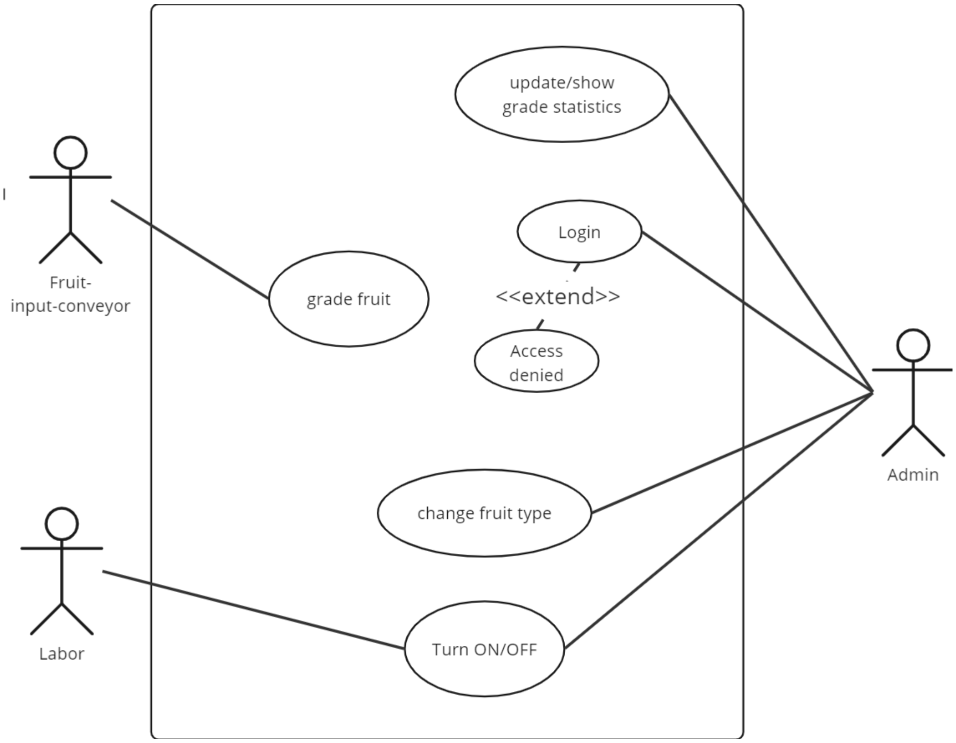

Figure 7. Use Case Diagram for Smart Fruit Grading System

Use cases:

7.1.1 Use Case 1 (login)

The "Login" use case involves the interaction of the labor with the system for the purpose of user authentication and session management. The main function is to allow a user to log into their account and subsequently log out to terminate the session securely.

Name	Description
Title	Login
Actor	Admin
Main Function	User authentication and session management
Brief Description	A user logs into his account and logs out to end the session.
Precondition	The user must have a registered account to use this functionality.
Flow of events summary	1.The user turns on the screen to log in 2. The user enters his username and password and initiates a log in request. 3. The system validates the credentials of the user.

	4. If the credentials are valid, the user will be logged in successfully; otherwise, the user will be prompted to renter their details.

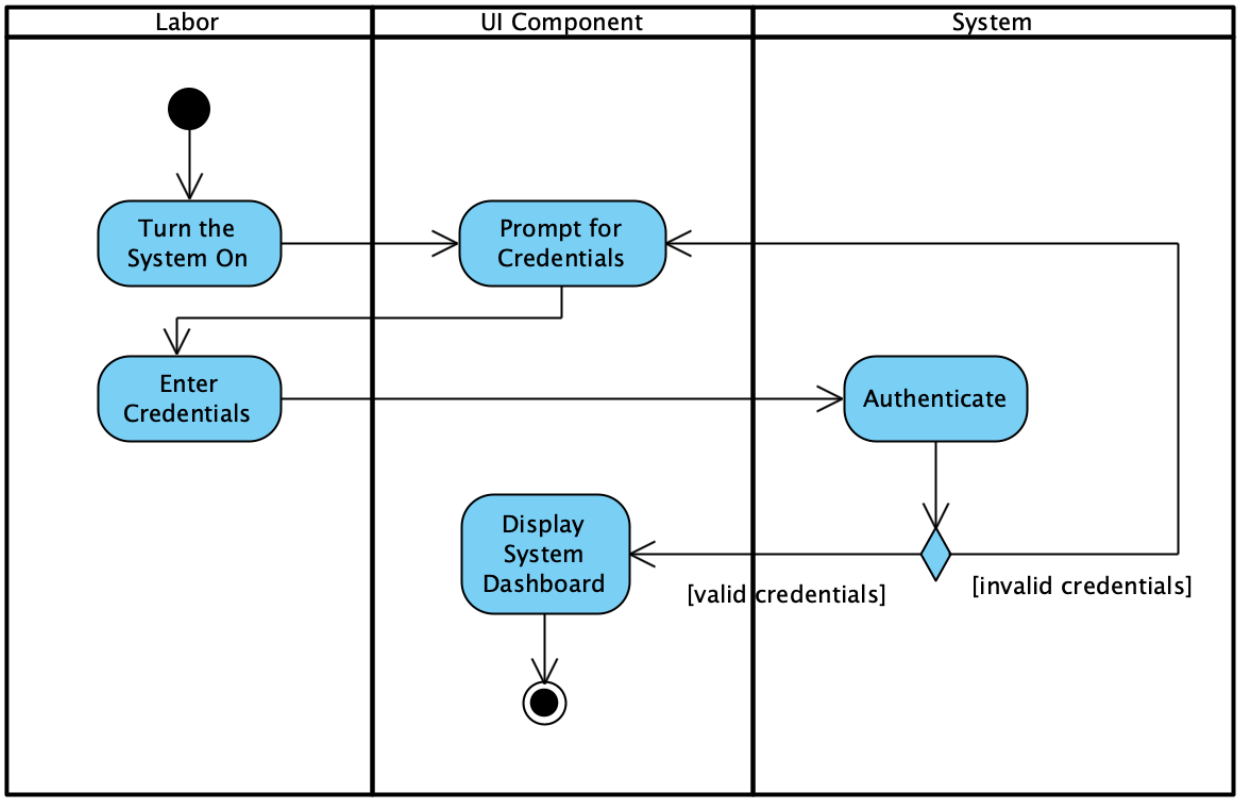

Figure 8. Activity Diagram of Login Use Case

7.1.2 Use Case 2 (Fruit Grading)

Name	Description
Title	Fruit Grading
Actor	Fruit-input-conveyor, Fuit-ouput-conveyor
Main Function	Grade the Fruits in 4-levels. The levels are High, Medium, low, and inedible
Brief Description	The fruits enter the system by input-gate then, it grades the fruit and assigns it on different lines depending on the grade assigned to the fruit.
Precondition	the system must be Turned ON.
Flow of events summary	1. Fruit Enter the system through input-gate 2. The fruit will continue forward due to the Fruit conveyor. 3. then the Fruit will trigger Image-capturing From the standard & and X-ray cameras by Infrared sensors.

	4. the AI module will grade the fruits depending on the Images taken. 5. GateInterface depending on the grade of the fruit will control the output-gate to make the fruit go to the corresponding line to its grade.

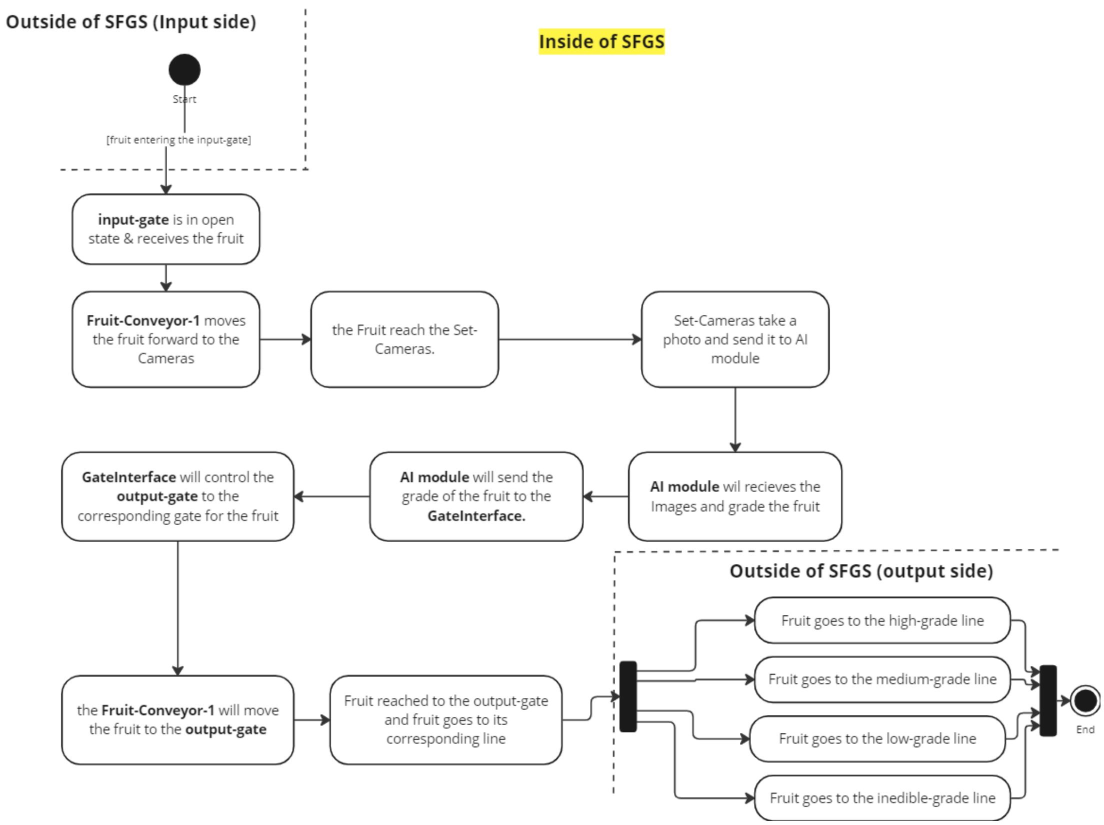

Figure 9. Activity Diagram of Fruit Grading

7.1.3 Use Case 3 (Turn ON/OFF the system)

Name	Description
Title	Turn ON/OFF
Actor	Labor,Admin
Main Function	The main idea is to turn the system ON and OFF
Brief Description	The usual functionality used to initiate the system.
Precondition	The labor has a convenient access to the system
Flow of events summary	1. The turn-ON-OFF switch will be toggled either to the ON or OFF position

	2. The Microcontroller will be by its hardware turn ON or OFF

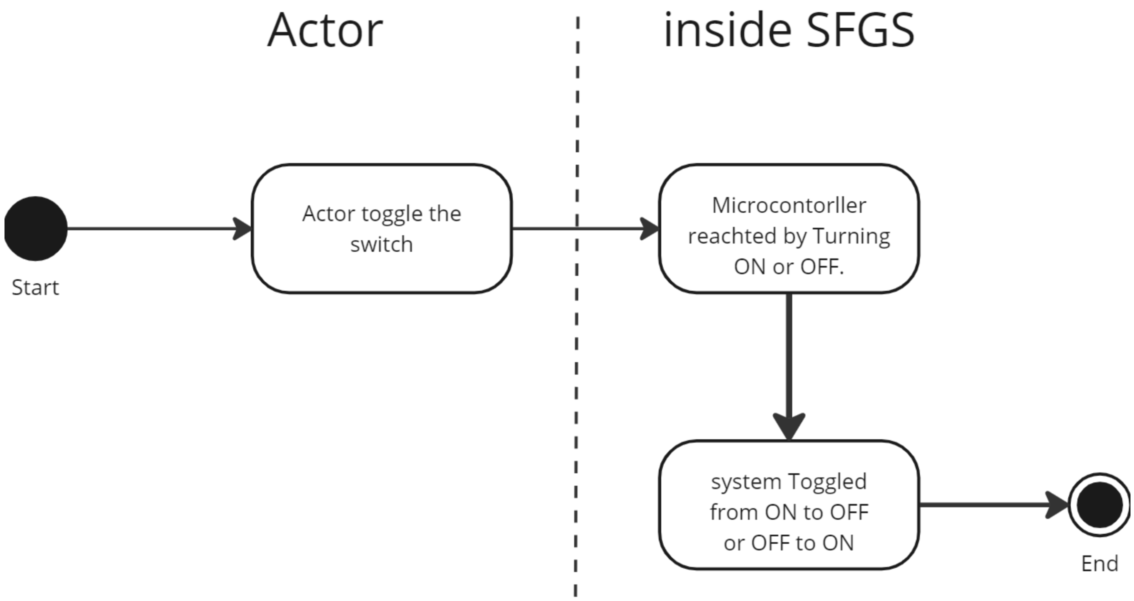

Figure 10. Activity Diagram of Turn ON/OFF

7.1.4 Use Case 4 (Set Fruit type)

Different types of fruits come into the factory in bulks: bulks of apples, oranges, and so on. The system expects one type of fruit at a time, and this type should be predetermined by the system manager. For example, when a truck of apples enters the system, we should change the type of fruit within the system to match the type being processed.

Name	Description
Title	Set Fruit type
Actor	Admin
Main Function	Set/change the fruit's type
Brief Description	Allows a user (A Labor) to change the type of a fruit through the UIComponent.
Precondition	The user must be logged in and have the necessary permissions to modify fruit types.
Flow of events summary	1. The user (A Labor) accesses the UIComponent to change the fruit type. 2. The UIComponent prompts the user to select a fruit and choose a new type. 3. The user selects the desired fruit and specifies the new type. 4. The system validates the request and updates the fruit type. 5. The UIComponent reflects the change in the displayed fruit type.

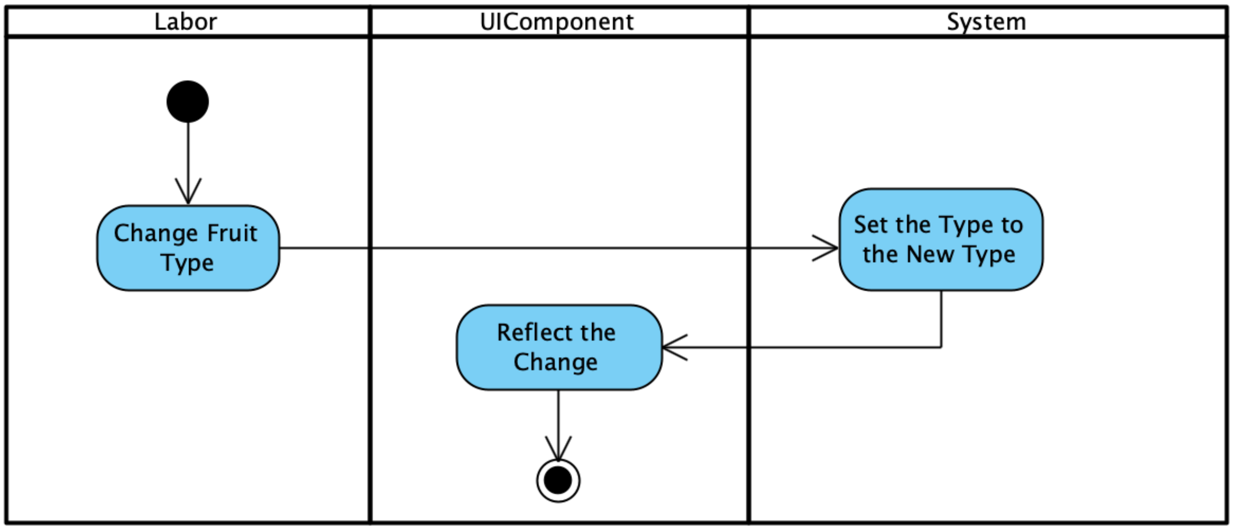

Figure 11. Activity Diagram of Set Fruit Type

7.1.5 Use Case 5 (Update/Show Grade Statistics)

This use case involves the AIModelComponent, which serves as an integral part of the system. It is responsible for maintaining and displaying statistics related to fruit types. The primary function of this use case is to update the running statistics of fruit types based on the class label provided by the AIModelComponent and reflect these updates in the user interface through the UIComponent.

Name	Description
Title	Update/show grade statistics
Actor	Admin
Main Function	Update the running statistics
Brief Description	The AIModelComponent provides the class label to update the number of fruit types
Precondition	The AIModelComponent is initialized and is in operation.
Flow of events summary	1. The AIModelComponent provides the class label to the system. 2. The system updates the current counters. 3. The update is reflected to the UIComponent to update the corresponding counter.

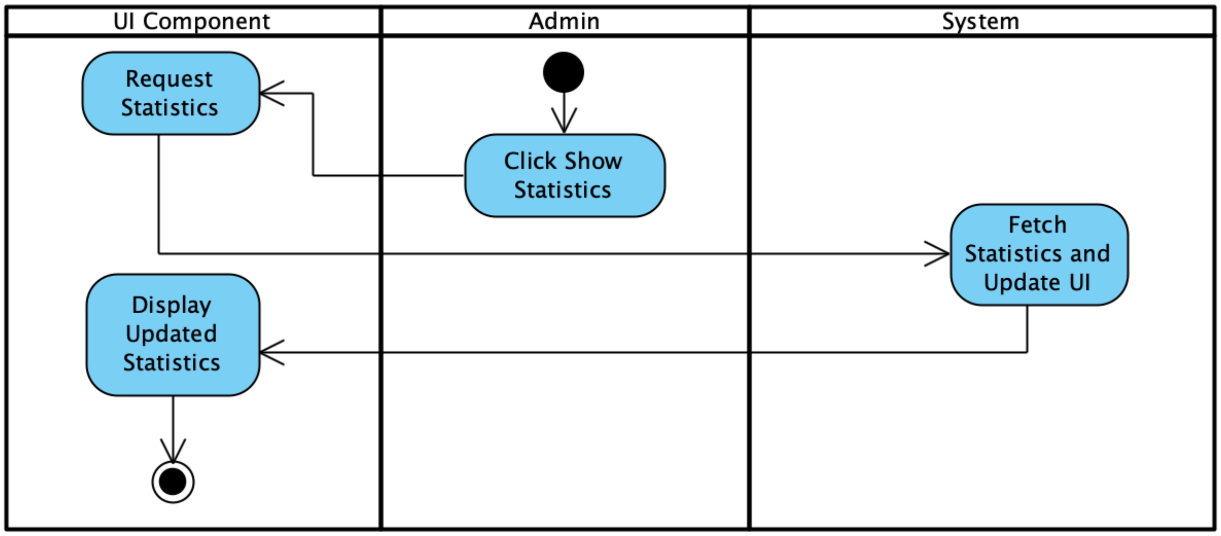

Figure 12. Activity Diagram for Update Use Case Illustrating the Interactions Among the Involved Components

7.2. Requirements

#	Requirement Description	Use Case Generated from
1	The system must provide a secure login mechanism for the users through the UIComponent.	Login
2	The system must provide a user friendly and intuitive login interface.	Login
3	The system must provide an option for account recovery thru a verified personal phone number.	Login
4	The system is required to maintain security by refraining from displaying both the password and username during the login process to mitigate the risk of social attacks (Shoulder Surfing)	Login
5	The system must efficiently process and apply the user's selection of a new fruit type, with the change reflected on the UIComponent within 500 milliseconds	Change Fruit Type
6	The system must prompt the user for a message to confirm if the user intended the change or not.	Change Fruit Type
7	Upon changing the fruit type, the system must display a confirmation message to the user, confirming the successful execution of the change	Change Fruit Type
8	Users should have the ability to customize the list of available fruit types through the UIComponent, allowing for personalized categorization	Change Fruit Type

9	The system must display real-time running statistics for different fruit types on the UIComponent and update them continuously as the system progresses.	Update/show grade statistics.
10	The update of statistics should be reflected simultaneously as each fruit progresses through the system.	Update/show grade statistics.
11	The system must visually represent grade statistics using charts or graphs on the UIComponent, providing users with a more intuitive understanding of the data	Update/show grade statistics.
12	The system must show the statistics in number of boxes where each box represents 50 units of fruit.	Update/show grade statistics.
13	The system must achieve a real-time fruit grading accuracy of at least 95% for a diverse set of input, ensuring reliable and precise classification.	Fruit grading
14	The time taken for a fruit to enter and exit the system, from the moment it is detected until the completion of grading, should not exceed 5 seconds to maintain efficiency.	Fruit grading
15	The system should be able to correctly direct each fruit to its designated line based on the assigned grade (High, Medium, Low, and Inedible).	Fruit Grading
16	The system should seamlessly integrate with an automated sorting mechanism to physically direct each graded fruit to its designated line without manual intervention	Fruit Grading
17	The switch for ON/OFF should be hard to toggle. It should need a grown human to toggle the switch	Turn ON/OFF the system
18	The switch should have a debouncer. To save the system form damage of bouncing.	Turn ON/OFF the system
19	turn-ON-OFF-switch should be easily reachable to the human when stands in front of the monitor and keyboard.	Turn ON/OFF the system
20	It's crucial for the turn-ON-OFF switch to clearly indicate its state. Make the ON and OFF symbols easily recognizable within the switch interface to improve user understanding.	Turn ON/OFF the system

7.3.Dynamic behavior of architecture

7.3.1. Sequence Diagram of Use Case 1 (Login)

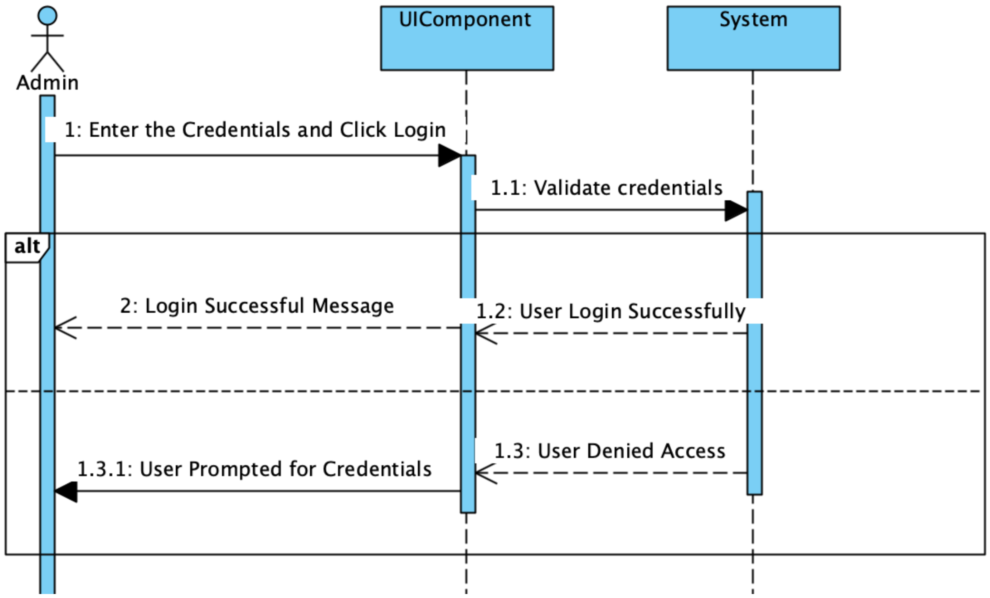

Figure 13. Login Sequence Diagram

The login sequence begins as the user provides their username and password through the application's interface. The system, managed by the LoginController, validates the input and initiates an authentication request to the AuthenticationService. This service queries the User Database to verify the credentials, responding with an authentication outcome. If valid, the LoginController grants access to the user's interface, transitioning to the main dashboard. Simultaneously, the LoggingService records the login attempt for auditing. In the case of invalid credentials, the system denies access, prompting the user to re-enter the correct information. This succinct process ensures secure and efficient user authentication.

7.3.2. Sequence Diagram of Use Case 2 (Fruit Grading)

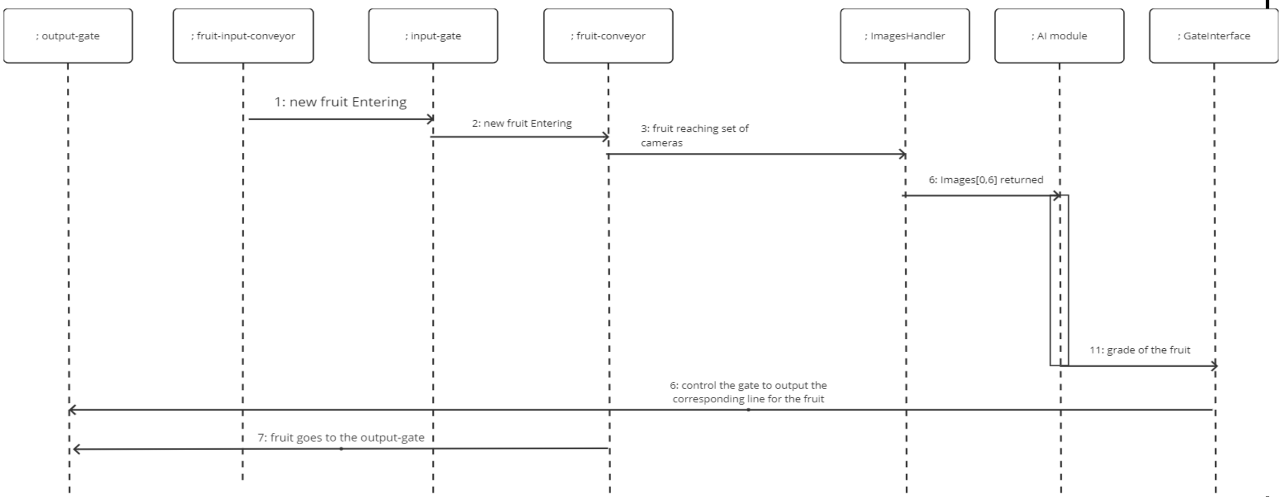

Figure 14. Sequence Diagram of Fruit Grading

7.3.3. Sequence Diagram of Use Case 3 (Turn ON/OFF the System)

Initially, the actor will toggle the switch, transitioning it from ON to OFF or vice versa. This switch's state directly influences a PIN in the microcontroller, determining the electrical status of the entire system—whether it is in the ON or OFF state. This functionality is implemented at the hardware level, a process further elucidated in Figure 15.

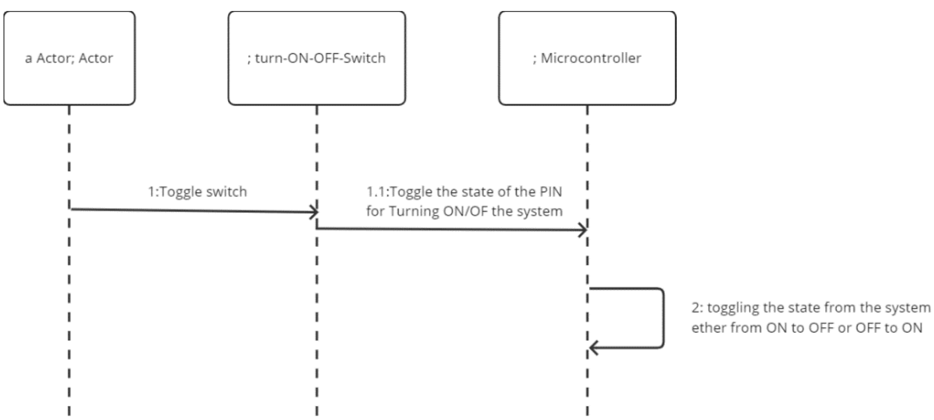

Figure 15 Sequence Diagram of Turn ON/OFF the System

7.3.4. Sequence Diagram of Use Case 4 (Set Fruit Type)

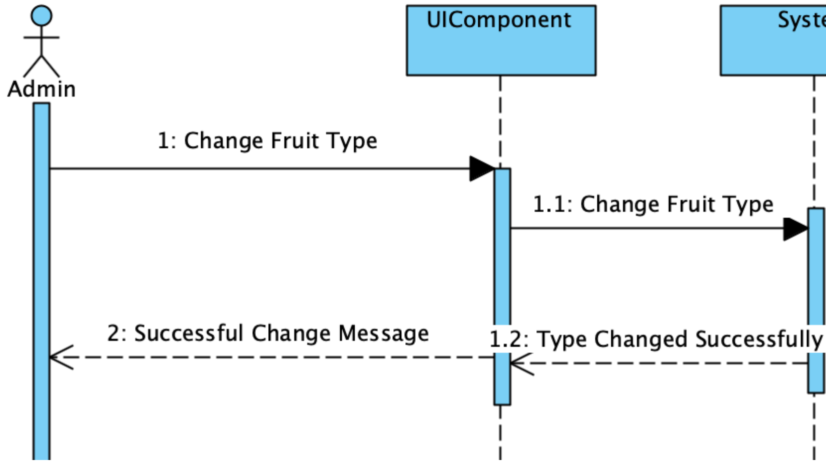

Figure 16. Sequence Diagram of Set Fruit Use Case

Within the system's architecture, meticulously crafted to meet functional requirements, the workflow for changing fruit types unfolds seamlessly. As the user (a laborer) interacts with the UIComponent to initiate the process, a well-defined sequence begins. The UIComponent prompts the user to choose a fruit and select a new type. Upon the user's selection, the system undertakes validation to ensure the legitimacy of the request. Once validated, the system updates the fruit type accordingly. The UIComponent, acting as the interface hub, promptly reflects this modification in the displayed fruit type, providing real-time feedback to the user. This carefully orchestrated workflow ensures a user-friendly and efficient process for modifying fruit types within the system, aligning with the overarching functional design of the architecture.

7.3.5. Sequence Diagram of Use Case 5 (Update/Show Grade Statistics)

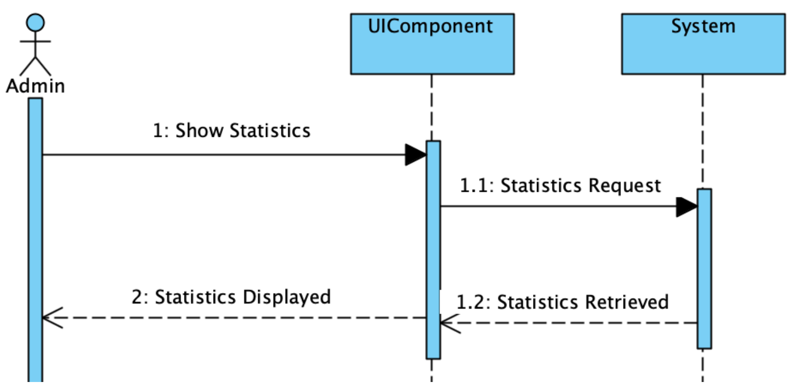

Figure 17. Sequence Diagram of Update/Show Statistics

In the system's architecture tailored to meet functional requirements, the process commences with the AIModelComponent supplying a class label to the system. This pivotal information triggers a sequence where the system promptly updates its current counters. The real-time update is seamlessly transmitted to the UIComponent, responsible for managing the user interface. This succinct and well-coordinated flow ensures that the UIComponent's corresponding counter is promptly and accurately updated. The carefully designed architecture allows for an efficient workflow, enabling different components to collaborate seamlessly in response to the system's main function.