Project Requirements Document (PRD)

for Color-Based Ball Collecting Robot

---

1. Introduction

This document defines the project requirements for a robot designed to collect balls based on specified colors. It outlines the functional and non-functional requirements, use cases, and acceptance criteria for successful implementation.

2. Business Requirements

The Color-Based Ball Collecting Robot aims to provide an efficient solution for sorting and collecting balls in various environments. The main objectives are to improve operational efficiency, enhance user experience, and ensure safety during operation.

3. System Overview

The robot will use sensors to detect the color of balls, navigate its environment, and collect the balls according to specified color criteria. The system will include:

- A microcontroller for processing data
- Color sensors for detecting ball colors
- Motors for movement
- A collection mechanism for gathering balls

---

4. Functional Requirements

- *Feature 1: User Interface*

1. Color Selection

- Users can select the color of the balls that the robot should collect from a predefined list (e.g., red, blue, green).

- *User Story*

Color Selection for Ball Collection

As a user, I want to select ball colors from a list so I can customize the robot's collection task.

- *Scenario*

Color Selection Process

- **Given** I'm on the color selection screen,
- **When** I view predefined colors and select one,
- **Then** the color is highlighted.
- **When** I click "Confirm,"
- **Then** my choice is registered, and the robot starts searching.
- **If no** balls are found,
- **Then** I receive a notification.

- *Acceptance Criteria*

1. Display Colors: Show a list of predefined colors.
2. Highlight Selection: Visually highlight the selected color.
3. Register Selection: Confirm selection on "Confirm" click.
4. Robot Feedback: Indicate when searching starts.
5. Notification: Alert if no balls are found.

- *Feature 2: Ball Detection*

1. Color Detection
- The robot should be able to detect the color of the balls using a color sensor.
2. Obstacle Avoidance
- The robot should navigate around obstacles using sensors to ensure that it can reach the balls effectively.

- *User Story*

Ball Detection for the Robot
As a user, I want the robot to detect ball colors and navigate obstacles to collect the desired balls effectively.

- *Scenario*

Ball Detection and Navigation

1. Color Detection

- **Given** the robot is on,
- **When** it approaches a ball,
- **Then** it detects the ball's color.
- **If** the color matches the selection,
- **Then** it identifies the ball.

2. Obstacle Avoidance

- **Given** the robot is moving towards a ball,
- **When** it encounters an obstacle,

- **Then** it detects the obstacle.
- **If** detected,
- **Then** it navigates around it and continues towards the ball.

- *Acceptance Criteria*

1. Color Detection: Accurately detect ball colors and identify matches.
2. Obstacle Avoidance: Detect obstacles and navigate around them without losing sight of the ball.

- *Feature 3: Ball Collection*

1. Collecting Mechanism
- The robot should have a mechanical arm or scoop that is capable of picking up the balls once they are detected.

2. Ball Storage
- The robot should store the collected balls in a designated compartment until the user retrieves them.

- *User Story*

Ball Collection by the Robot
As a user, I want the robot to collect and store balls for easy retrieval later.

- *Scenario*

Ball Collection Process

1. Collecting Mechanism

- **Given** the robot detects a ball,
- **When** it positions its arm above it,
- **Then** it picks up the ball.

Outcome: The ball is collected.

2. Ball Storage

- **Given** the ball is collected,
- **When** the robot transports it,
- **Then** it stores the ball in the compartment.

Outcome: The ball is stored until retrieval.

- *Acceptance Criteria*

1. Collecting Mechanism: The robot must pick up balls reliably.
2. Ball Storage: The robot must have a secure compartment for collected balls.

- *Feature 4: Navigation*

1. Path Planning

- The robot should be able to plan a path to the detected balls using algorithms such as A* or Dijkstra.

2. Motion Control

- The robot should support movement in multiple directions (forward, backward, left, right) to navigate the collection area.

- *User Story*

Navigation for Ball Collection Robot

As a user, I want the robot to navigate efficiently to detected balls for smooth collection.

- *Scenario*

Navigation to Detected Balls

1. Path Planning:

- **Given** the robot detects a ball,
- **When** it calculates the route,
- **Then** it uses algorithms (e.g., A* or Dijkstra) for the optimal path.

Outcome: The robot finds the best route to the ball.

2. Motion Control:

- **Given** the planned path,
- **When** the robot moves,
- **Then** it navigates in multiple directions (forward, backward, left, right).

Outcome: The robot reaches the ball while avoiding obstacles.

- *Acceptance Criteria*

1. Path Planning: Calculate the optimal path considering obstacles.

2. Motion Control: Support movement in multiple directions for effective navigation.

Feature 5: Feedback System

1. Status Indicators

- The robot should provide visual or auditory feedback to indicate when it has successfully collected a ball or encountered an error.

2. User Notifications

- The system should notify users via the interface when collection is complete or if any problems arise during operation.

- *User Story*

Feedback System for Ball Collecting Robot

As a user, I want the robot to provide feedback on its status and notify me of any issues to stay informed about the collection process.

- *Scenario*

Feedback and Notifications

1. Status Indicators:

- **Given** the robot has collected a ball,
- **When** the collection is complete,
- **Then** it gives visual (e.g., LED) or auditory (e.g., beep) feedback.

Outcome: User is confirmed of the collection.

- **Given** the robot encounters an error,
- **When** the error occurs,
- **Then** it provides a different signal.

Outcome: User is alerted to the issue.

2. User Notifications:

- **Given** the collection is complete,
- **When** all balls are collected,
- **Then** the system notifies the user via the interface.

Outcome: User is informed of completion.

- **Given** there is an operational problem,
- **When** an issue arises,
- **Then** the system provides a notification with details.

Outcome: User receives troubleshooting information.

- *Acceptance Criteria*

1. Status Indicators: Clear feedback for success and distinct error signals.
2. User Notifications: Notify upon completion and alert to issues with details.

---

5. Non-Functional Requirements

5.1 Performance
- Response Time: Process color detection and initiate collection within 5 seconds.
- Throughput: Collect multiple balls efficiently without delays.

5.2 Security
- Safety: Detect humans and obstacles for safe operation.

5.3 Usability
- User Interface: Intuitive design for easy color selection and status monitoring.
- Documentation: Provide clear operating instructions.

5.4 Reliability
- **Uptime:** Minimum 95% operational uptime.
- **Error Recovery:** Graceful recovery from errors without user intervention.

---

6. Use Cases

- *Use Case 1: Selecting Ball Color*

- **Actors:** User
- **Description:** The user selects a specific color for the robot to collect.
- **Preconditions:**
  • The robot is powered on.
  • The user interface is accessible.
- **Steps:**
  1. The user opens the interface.
  2. The system displays the available color options (e.g., red, blue, green).
  3. The user selects the desired color.
  4. The system confirms the selection and activates the robot.
- **Postconditions:**
  • The robot is configured to collect balls of the specified color.

- *Use Case 2: Detecting Ball Colors*

- **Actors:** Robot
- **Description:** The robot identifies balls of the specified color using its sensors.
- **Preconditions:**
  • The user has selected a target color.
  • The robot is in operation mode.
- **Steps:**
  1. The robot scans the area using color sensors.
  2. The system identifies balls of the specified color.
  3. If a ball matches the color criteria, the robot prepares to collect it.
- **Postconditions:**
  • The system flags the detected ball for collection.

- *Use Case 3: Avoiding Obstacles*

- **Actors:** Robot
- **Description:** The robot navigates around obstacles in its path.
- **Preconditions:**
  • The robot is in movement mode.
  • Obstacles are present in the operating area.
- **Steps:**
  1. The robot uses distance sensors to detect obstacles.
  2. The system calculates an alternate path to avoid collisions.
  3. The robot resumes its operation once the obstacle is cleared.
- **Postconditions:**

• The robot successfully avoids the obstacle without user intervention.

- *Use Case 4: Collecting Balls*

- Actors: Robot
- Description: The robot uses a mechanical mechanism to collect detected balls.
- Preconditions:
• A target ball has been detected.
• The collection mechanism is functional.
- Steps:
1. The robot approaches the ball.
2. The collection mechanism picks up the ball.
3. The ball is stored in the designated compartment.
4. The system confirms successful collection.
- Postconditions:
• The ball is securely stored, and the robot continues its task.

- *Use Case 5: Providing Status Updates*

- Actors: Robot, User
- Description: The robot provides feedback on its current status.
Preconditions:
• The robot is in operation.
• A feedback system (e.g., LEDs, LCD, or buzzer) is installed.
- Steps:
1. The robot completes a task (e.g., collecting a ball or encountering an error).
2. The system sends status updates to the user interface.
3. The user reviews the feedback to monitor progress.
- Postconditions:
• The user is informed of the robot's activities and potential issues.

---

7. Constraints

- Component Limitation: Use only built-in components compatible with the robot's design.
- Software Compatibility: Ensure all sensors and components work seamlessly with the chosen microcontroller.

---

8. Timeline

- Complete implementation within 9-10 weeks.

---

9. Priority of Features

- **Must Have**

- Detect and collect specified color balls.
- Obstacle avoidance.
- Basic movement capabilities.

- **Should Have**

- Display current status on an LCD.
- More advanced navigation features.

- **Could Have**

- Use a camera for advanced color detection.
- Implement machine learning for improved navigation.

- **Won't Have**

- Complex mapping or extensive data logging (future enhancement).

---

10. Glossary

- Feature: A high-level system capability. Example: "Detect and collect balls of a specific color."
- User Story: A description of a user's need. Example: "As a user, I want the robot to collect only red balls so that I can organize them efficiently."
- Scenario: Describes different ways a user interacts with the system. Example: Successful ball collection, obstacle in the robot's path, or color detection failure.
- Acceptance Criteria: A set of conditions that determine when a user story is complete. Example: "Given the robot detects a red ball, it should successfully collect it and notify the user."
- Functional Requirement: Describes specific features and actions the system must support. Example: "The robot must detect the color of balls using sensors and collect them with a mechanical arm."
- Non-Functional Requirement: Describes the system's quality attributes, like speed, security, and usability. Example: "The robot must detect and collect a ball within 5 seconds of identifying it."

Implementation plan

Week 3	***Define project requirements and***

	prepare components
Week 4	*Physically assemble the robot and test the core components*
Week 5	*Develop the color detection software and ball collection system*
Week 6	*Program the guidance system, obstacle avoidance and system integration*
Week 7	*Add the user interface and test the entire robot*
Week 8	*Optimize performance and conduct final tests*
Week9	*Final project delivery*