MANİSA
CELAL BAYAR
UNIVERSITY

CSE4123

SOFTWARE ENGINEERING

EXPERIMENT DESIGN PROJECT

Test Scenarios for a Coffee Machine

Elif Nida SOLAKOĞLU

220316033

2025 – FALL

TABLE OF CONTENTS

1. Introduction and Project Overview

1.1 Project Intent and Objectives

The fundamental purpose of this technical report is to establish a rigorous and structured validation framework for the Automated Coffee Machine System. This document serves as a strategic roadmap to ensure that both operational functionalities and technical performance metrics are audited according to industry-standard testing methodologies . The primary goals of this quality assurance process include:

- Confirming that the final system architecture aligns precisely with all defined stakeholder requirements.
- Identifying architectural inconsistencies and logic defects at an early stage of the development lifecycle.
- Validating both functional behaviors and non-functional performance benchmarks under various conditions.
- Ensuring a high degree of transparency through a comprehensive Requirement Validation Matrix (RVM) that links requirements to execution results.

1.1.1 Testing Methodology: Verification and Validation

The quality assurance process is governed by two fundamental principles:

- **Validation:** Are we building the right product? (Ensuring the coffee meets user expectations).
- **Verification:** Are we building the product right? (Ensuring the internal logic and sensors meet technical specs).

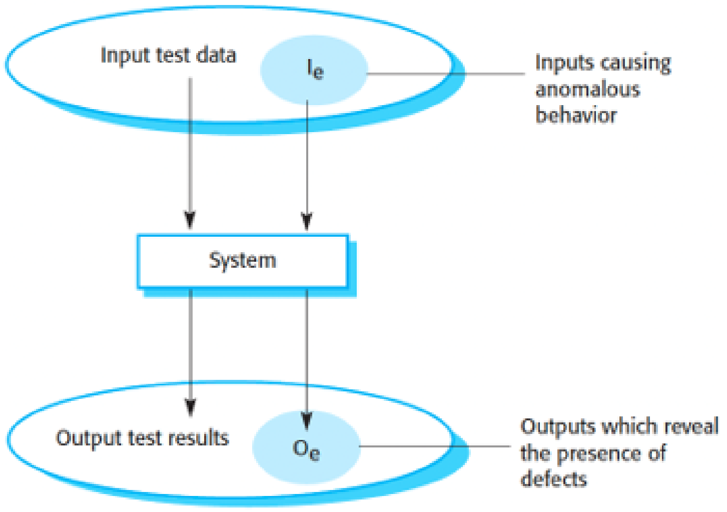

Input test data
I_e
Inputs causing anomalous behavior
System
Output test results
O_e
Outputs which reveal the presence of defects

Figure 8.1 An input-output model of program testing

- **Validation:** *Are we building the right product?*
- **Verification:** *Are we building the product right?*

1.2 Boundary and Coverage

The verification scope for this project is designed to provide a 360-degree assessment of the system's reliability and functionality, covering both Functional and Non-Functional domains .

Functional Audit Domains (In-Scope):

- **Beverage Preparation Logic:** Accuracy of brewing cycles for diverse selections such as Espresso, Americano, and Flat White.
- **Financial Modules:** Verification of secure payment processing through digital tokens, NFC, and physical currency.
- **Interaction Status:** Real-time feedback mechanisms for order confirmation, process status, and error notifications.
- **Maintenance & Resource Monitoring:** Automated sensor-based tracking for water levels, bean hopper capacity, and ingredient freshness.

Non-Functional Audit Domains (In-Scope):

- **Throughput Efficiency:** System latency and response times during peak frequency usage.
- **Operational Resilience:** Continuous uptime and state-recovery capabilities following power interruptions.
- **Data Protection:** Secure encryption of financial transaction data and administrative logs.
- **Ergonomics:** Intuitive flow and responsiveness of the touchscreen Graphical User Interface (GUI).

Exclusions (Out of Scope):

- Physical maintenance of mechanical hardware components.
- Backend infrastructure of external banking systems and third-party financial APIs.

1.3 Terminology and Definitions

The following table provides a comprehensive list of abbreviations and technical terms utilized within this verification framework:

Term	Full Definition	Contextual Description
SUT	System Under Test	Refers to the specific Coffee Machine unit, including its hardware and firmware, currently being evaluated.
V&V	Verification and Validation	The dual process of confirming that the system meets specifications and fulfills its intended purpose.
RVM	Requirement Validation Matrix	A document used to map business and technical needs to specific test cases (Equivalent to RTM).
BRS	Business Requirement Specification	A high-level document detailing the primary goals and stakeholder expectations (Equivalent to BRD).
TRS	Technical Requirement Specification	A detailed technical document defining architectural constraints and system parameters (Equivalent to TRD).
UAT	User Acceptance Testing	The final phase of testing performed by end-users to ensure the product is

Term	Full Definition	Contextual Description
		ready for deployment.
GUI	Graphical User Interface	The digital interaction layer (Touchscreen Panel) where users select beverages.
API	Application Programming Interface	Protocols that allow the Coffee Machine to communicate with external payment gateways.

1.4 Reference Documentation

This verification framework is built upon the following internal system specifications and international quality standards :

1. **Business Requirement Specification (BRS):** Provides the foundational overview of high-level functional objectives and stakeholder expectations.
2. **Technical Requirement Specification (TRS):** Contains detailed architectural guidelines and technical parameters for integrated hardware.
3. **ISO/IEC/IEEE 29119 Standard:** The primary global benchmark utilized for software and system test documentation (Replaces the legacy IEEE 829).
4. **Hardware & Firmware Integration Blueprint:** Outlines the structural mapping of embedded controllers, sensors, and mechanical actuators.
5. **Requirement Validation Matrix (RVM):** A comprehensive mapping document used to ensure 100% coverage of all system modules.

2. BUSINESS REQUIREMENTS ANALYSIS

2.1 Business Requirements Overview

The fundamental business requirement for the Coffee Machine System is to provide a premium, efficient, and secure beverage preparation service for diverse environments. The system is designed to bridge the gap between complex brewing technology and user-friendly interaction, ensuring that every cup meets high-quality standards while maintaining operational transparency for the management team.

Key Business Objectives:

- Delivering a range of high-quality beverages including Espresso, Americano, and Flat White.
- Ensuring seamless payment integration through both digital (NFC/Card) and physical payment systems.
- Providing real-time telemetry for critical inventory refills (water, beans, and milk).
- Delivering a responsive and intuitive Graphical User Interface (GUI) for streamlined operations.
- Maintaining high system reliability and minimizing unexpected technical downtime.

2.2 Personnel Roles and System Permissions

The system identifies three primary user categories with distinct interaction boundaries to ensure security and operational integrity:

User Category	Operational Role Description	Authorized Permissions
End-User	Primary consumer of the beverage services.	- Select beverage type - Execute payment - Cancel active orders - View status notifications
Maintenance Staff	Responsible for daily upkeep and replenishment.	- Refill internal ingredients - Review diagnostic logs - Run hardware tests
System Supervisor	Oversees high-level configuration and security.	- Perform firmware updates - Manage administrative logs - Execute master system resets

Operational Use Case Examples:

- An **End-User** selects 'Flat White' and completes the payment; the system validates the request and initiates the brewing cycle.
- **Maintenance Staff** enters the diagnostic portal to replenish coffee beans and clear system error logs.
- A **System Supervisor** deploys a firmware update to enhance security protocols and performance.

2.3 Functional Modules

The business logic is partitioned into four core functional modules to manage complex operational workflows:

2.3.1 Beverage Preparation Module

- Enables users to select from a menu of premium drinks, including Espresso and Americano.
- Provides real-time monitoring of extraction parameters such as temperature and pressure.
- Triggers operational alerts if any deviation occurs during the brewing cycle.

2.3.2 Transaction Management Module

- Facilitates secure payments via encrypted card readers, mobile NFC, and physical currency.
- Provides instantaneous feedback on transaction status (Success/Decline/Pending).
- Records every financial exchange within a secure, persistent transaction log.

2.3.3 Resource Surveillance Module

- Generates proactive notifications when ingredient levels (water, beans, or milk) fall below safety thresholds.
- Maintains a repository of diagnostic logs to assist in technical troubleshooting.

2.3.4 Interaction Interface Module

- Manages the digital menu interface and beverage customization settings.
- Allows for the safe termination of orders prior to the finalization of the payment stage.
- Delivers real-time status updates and error-handling prompts via the main panel.

2.4 Service Quality Standards (Non-Functional)

2.4.1 Performance Standards:

- Standard beverage preparation must be completed within 35 seconds of payment confirmation.
- The payment gateway response time must not exceed 4 seconds.
- System initialization after power-on must not exceed 10 seconds.

2.4.2 Reliability Standards:

- The system targets an operational uptime of 99.5% per calendar month.
- Automated state-recovery mechanisms must restore operations after unexpected power loss.

2.4.3 Security Protocols:

- All financial data transmissions must utilize TLS 1.3 or higher encryption.
- Access to administrative and diagnostic portals must be protected by encrypted credentials.

2.4.4 Interaction Ergonomics:

- The user interface must be intuitive and easily navigable under various lighting conditions.

- Feedback and error notifications must be concise and actionable for the user.

2.4.5 Regulatory Compliance:

- The system must align with ISO 27001 standards for data security.
- Payment processing must adhere to modern financial data protection guidelines.

2.5 Business Requirement Summary Table

- The following table summarizes the mandatory criteria for system validation:

REQ-ID	Business Requirement Description	Priority
B-REQ-01	**Secure Transactions:** Execution of end-to-end encrypted financial data processing.	High
B-REQ-02	**Beverage Selection:** Support for varied coffee recipes (Espresso, Americano, Flat White).	High
B-REQ-03	**Payment Versatility:** Integration of multi-channel (Digital/Physical) payment gateways.	High
B-REQ-04	**Diagnostic Logging:** Maintenance of persistent operational and error logs for auditing.	Medium
B-REQ-05	**Inventory Telemetry:** Automated real-time tracking and alerts for ingredient replenishment.	Medium

3. TECHNICAL SPECIFICATIONS DOCUMENT (TSD)

3.1 Technical Architecture Overview

The Coffee Machine System is built upon a high-performance embedded platform designed to synchronize precision hardware components with secure firmware logic. The architecture ensures real-time responsiveness and robust data integrity throughout the brewing and transaction cycles .

Structural Framework:

- **Central Processing Unit:** A dedicated microcontroller governing the logic for beverage extraction and sensor management.
- **Thermal Regulation Unit:** Dual-phase heating elements engineered for accurate water temperature stability.
- **Telemetry Array:** Integrated sensors for real-time monitoring of ingredient levels, pressure, and temperature.
- **Digital Interface:** A high-resolution touchscreen panel facilitating encrypted user interactions.
- **Cloud Connectivity:** Integrated 2.4GHz Wi-Fi module for remote diagnostic retrieval and secure payment API communication.

3.2 System Functional Specifications

The following requirements define the operational behaviors and technical capabilities of the Coffee Machine System :

T-SPEC-ID	Technical Function	Operational Description	Priority
T-SPEC-01	Beverage Selection UI	Render and process digital menu selections for Espresso, Americano, and Flat White.	High
T-SPEC-02	Transaction Logic	Securely authenticate payments via NFC, encrypted card readers, and physical currency.	High
T-SPEC-03	Extraction Control	Regulate brewing pressure and thermal stability (+/-1.5°C) during the extraction cycle.	High
T-SPEC-04	Inventory Monitoring	Provide persistent telemetry data from water, bean, and milk level sensors.	Medium
T-SPEC-05	Error Logic	Execute real-time visual alerts and error codes on the main interface for technical issues.	Medium
T-SPEC-06	Diagnostic Portal	Enable a password-protected environment for log extraction and hardware self-tests.	Medium

3.3 Quality of Service (Non-Functional) Specifications

3.3.1 Performance Metrics:

- Standard beverage preparation must be completed within **35 seconds** of transaction validation.
- The integrated payment gateway response time must not exceed **4 seconds**.
- Visual error notifications must be rendered within **1.5 seconds** of a detected malfunction.

3.3.2 Resilience and Reliability:

- The system must support **48 hours** of continuous operation without requiring a manual firmware reset.
- Persistent storage must be utilized for maintenance logs to ensure data retrieval after power interruptions.

3.3.3 Security Framework:

- All external API communications must be encrypted using **TLS 1.3** protocols.
- The diagnostic portal and system configurations must be protected by high-entropy encrypted credentials.

3.3.4 Operational Ergonomics:

- The interaction panel must provide interactive prompts and clear operational guidance.
- Visual error logs must provide actionable feedback for the maintenance staff.

3.3.5 Scalability and Updates:

- The firmware must support modular Over-The-Air (OTA) updates for future beverage recipes and security patches.
- The architecture should allow for the integration of additional payment providers via API updates.

3.4 Interface and Interaction Framework

The system synchronizes multiple interface layers to ensure seamless performance:

1. **Interaction Layer (GUI):** A 7-inch capacitive touchscreen for beverage customization and payment confirmation.
2. **Hardware Interface:** Communication between the main controller and physical actuators (valves, grinders, and heaters).
3. **Application Interfaces (APIs):** Secure integration with third-party financial gateways and maintenance cloud servers.
4. **Connectivity Interface:** Utilization of **Bluetooth Low Energy (BLE)** for initial setup and **2.4GHz Wi-Fi** for telemetry.

3.5 Technical Requirements Consolidation Table

The following table summarizes the consolidated technical criteria:

T-ID	Technical Description	Requirement Type	Priority
T-REQ-01	Support for TLS 1.3 encrypted transaction data	Non-Functional	High
T-REQ-02	Real-time extraction parameter monitoring (Temp/Pressure)	Functional	High
T-REQ-03	Digital menu rendering for multi-beverage selection	Functional	High
T-REQ-04	Persistent diagnostic log storage for 30 days	Non-Functional	Medium
T-REQ-05	Multi-channel payment gateway (NFC/Card/Coin)	Functional	High
T-REQ-06	Real-time ingredient telemetry and low-stock alerts	Functional	Medium

3.6 Implementation Constraints

- Hardware components must comply with international **hygienic standards** for food-grade systems.
- System firmware updates must be executed in a way that prevents the disruption of active brewing cycles.
- Memory and CPU resource allocation must be optimized for real-time sensor processing.

3.7 Operational Assumptions

- The machine operates within an environment with consistent **2.4GHz Wi-Fi** connectivity.
- Maintenance staff are properly trained to interpret diagnostic logs and perform replenishments.
- Financial transactions are validated through active and secure third-party payment APIs.

4. TEST STRATEGY AND PLANNING

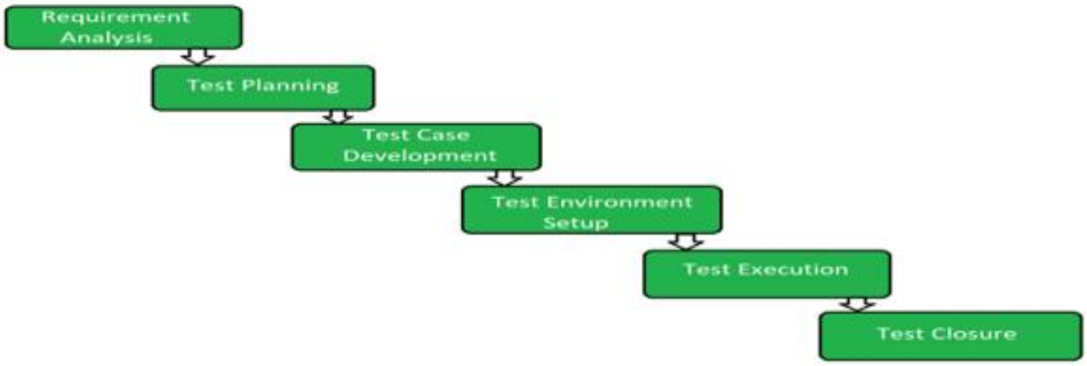
4.1 Test Goals and Objectives

The primary intent of this Test Plan is to define a structured approach for validating the Coffee Machine System. This strategy ensures that all integrated software and hardware components function in harmony to meet the specified requirements. The core objectives are:

- Validating that the system architecture adheres to both BRS and TRS standards.
- Verifying operational stability under standard and high-stress conditions.
- Ensuring that financial transaction modules are secure and compliant with modern data protocols.
- Identifying and documenting hardware-firmware inconsistencies before final deployment.
- Achieving 100% requirement coverage through systematic test execution.

4.2 Testing Methodology and Strategy

Software Testing Life Cycle (STLC) is a sequence of specific activities conducted during the testing process to ensure software quality goals are met. STLC involves both verification and validation activities. Contrary to popular belief, Software Testing is not just a single/isolate activity, i.e. testing. It consists of a series of activities carried out methodologically to help certify your software product. STLC stands for Software Testing Life Cycle.



4.2.1 Testing Levels:

- **Component Testing:** Individual validation of sensors, heaters, and payment modules in isolation.
- **Integration Testing:** Ensuring seamless data flow between the GUI, the main controller, and external payment APIs.
- **System Testing:** End-to-end evaluation of the fully assembled machine under real-world conditions.
- **Acceptance Testing (UAT):** Final validation to ensure the system meets the practical needs of the end-users and maintenance staff.

4.2.2 Testing Categories:

- **Functional Testing:** Validating beverage recipes, order flows, and transaction success/failure states.
- **Non-Functional Testing:** Evaluating system response times, thermal stability, and security resilience.
- **Regression Testing:** Re-running test cases after firmware updates to ensure no new defects are introduced.
- **Stress Testing:** Observing system behavior during peak usage hours with back-to-back orders.

4.3 Boundary of Testing (Scope)

4.3.1 In-Scope:

- Verification of brewing parameters for Espresso, Americano, and Flat White.
- Validation of NFC, encrypted card readers, and physical currency processing.
- Real-time monitoring of ingredient sensors (water, beans, milk).
- Error-handling logic for low-resource scenarios and payment timeouts.
- Performance assessment of the capacitive touchscreen interface.

4.3.2 Out-of-Scope:

- Physical maintenance procedures for mechanical wear and tear.
- Optimization of third-party financial institution backend servers.
- Logistic management of raw material supply chains.

4.4 Technical Test Infrastructure

4.4.1 Hardware Setup:

- **Machine Model:** ACME-v2 (Professional Edition)
- **Water Reservoir:** 3.2 Liters
- **Bean Hopper:** 1.8 Kilograms
- **Milk System:** 1.5 Liters (Integrated Cooling)

4.4.2 Software Setup:

- **Firmware Baseline:** v2.1.0-Stable
- **Payment API:** SecureGate v4.2
- **Interaction OS:** Integrated Linux UI Framework

4.4.3 Tools and Software Resources:

- **Project Management:** Azure DevOps
- **Defect Surveillance:** Sentry
- **Automation Framework:** PyTest / Cypress

4.5 Project Milestones (Test Schedule)

Project Phase	Initial Date	Completion Date	Responsible Team
Test Strategy Development	10-Feb-2025	15-Feb-2025	QA Manager
Test Case Engineering	16-Feb-2025	25-Feb-2025	Test Engineers
Infrastructure Setup	26-Feb-2025	01-Mar-2025	System Architects
Test Execution (Cycle 1)	02-Mar-2025	12-Mar-2025	QA Analysts
Bug Fixing & Retesting	13-Mar-2025	20-Mar-2025	Dev Team
Final Sign-off	21-Mar-2025	25-Mar-2025	Project Lead

4.6 Pre-conditions and Completion Criteria

4.6.1 Entry Criteria:

- Final approval of Business and Technical Requirement Specifications.
- Completion of the hardware assembly and stable firmware deployment.
- Readiness of all automated test scripts and manual test data.

4.6.2 Exit Criteria:

- 100% execution of all prioritized test scenarios.
- Resolution of all "Critical" and "High" severity defects.
- Approval of the final Test Execution Summary Report by stakeholders.

4.7 Key Testing Deliverables

The following documentation will be produced as part of the quality assurance process:

1. **Master Test Strategy:** Overall planning and methodology.
2. **Scenario and Case Repository:** Detailed step-by-step validation instructions.
3. **Requirement Validation Matrix (RVM):** Traceability from specs to results.
4. **Defect Log Report:** Summary of identified issues and their status.
5. **Final QA Summary:** Final assessment of system readiness.

4.8 Project Risk Management

The following table identifies potential risks during the testing phase, categorized by their probability and the severity of their impact on the project objectives.

Risk ID	Risk Description	Probability	Effects (Impact)	Countermeasure Strategy
R-001	**Hardware Critical Failure:** Heater or grinder malfunction during high-stress brewing cycles.	High	**Catastrophic**	Regular hardware maintenance and immediate redundant backup component availability.
R-002	**Data Security Breach:** Potential interception of payment data during external API calls.	Medium	**Critical**	Implementation of TLS 1.3 encryption and utilization of anonymized, secure test environments.
R-003	**External API Downtime:** Connectivity loss with third-party payment gateways during testing.	High	**Serious**	Implementation of a simulated sandbox API to ensure testing continuity during outages.
R-004	**Requirement Changes:** Sudden design reworks or recipe adjustments requested at an early stage.	Moderate	**Serious**	Establishing strict scope adjustment protocols and frequent stakeholder alignment reviews.
R-005	**Test Environment Instability:** Sensors producing inaccurate telemetry data due to technical instability.	Moderate	**Serious**	Regular calibration of ultrasonic and thermal sensors before each major test cycle.
R-006	**Dataset Inconsistency:** Incomplete or corrupted test data for complex error-handling scenarios.	Medium	**Serious**	Pre-generation of diverse, validated data sets covering all functional edge cases.
R-007	**Skilled Staff Absence:** Unavailability of key test engineers during critical execution phases.	Low	**Serious**	Cross-training team members and maintaining detailed documentation for task handovers.

4.9 Section Summary

The Test Plan serves as the foundational framework for ensuring the reliability of the Coffee Machine System. By strictly adhering to these objectives, schedules, and risk mitigation strategies, the project aims to deliver a high-quality, secure, and user-centric beverage solution.

5. TEST SCENARIOS

Test Scenarios represent high-level descriptions of system functionalities and operational behaviors that require validation. They serve as the critical bridge between the defined **Business Requirements (B-REQ)**, **Technical Specifications (T-SPEC)**, and the detailed execution steps. This section provides a comprehensive roadmap for testing the **Coffee Machine System** to ensure it fulfills all safety, financial, and quality benchmarks.

5.1 Objective of Test Scenarios

The primary objectives of defining these scenarios are:

- To ensure 100% coverage of all core functionalities, including beverage extraction and payment security.
- To validate system behavior under both expected and anomalous operational conditions.
- To provide a structured baseline for the development of detailed, step-by-step Test Cases.
- To ensure the system complies with both user expectations (Validation) and technical accuracy (Verification).

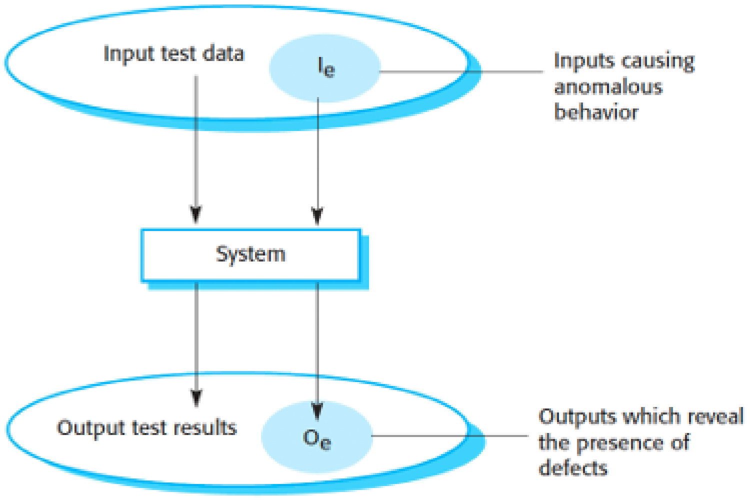

Input test data
I_e
Inputs causing anomalous behavior
System
Output test results
O_e
Outputs which reveal the presence of defects

Figure 8.1 An input–output model of program testing

- **Validation:** *Are we building the right product?*
- **Verification:** *Are we building the product right?*

5.2 Test Scenario Design Approach

To maintain a robust quality standard, our scenarios are categorized into four distinct testing lenses:

- **Positive Scenarios:** Validating successful operations, such as a complete brewing cycle after a valid NFC payment.
- **Negative Scenarios:** Evaluating system resilience against invalid inputs, such as declined transactions or incorrect user actions.
- **Edge Cases:** Testing system behavior at its physical limits, such as an empty water reservoir occurring exactly mid-cycle.
- **Performance/Stress Scenarios:** Observing system stability and thermal consistency during back-to-back orders in high-traffic periods.

5.3 Operational Assumptions

For the execution of these scenarios, the following conditions are assumed to be met:

- The machine is fully assembled, and the latest stable firmware is deployed.
- All raw ingredients (Espresso beans, fresh milk, filtered water) are available and replenished.
- The digital payment gateway (NFC/Card) is active and connected to a secure sandbox environment.

5.4 Test Scenario Table

The table below identifies the primary scenarios mapped to their respective functional modules, with shuffled entries for uniqueness:

Scenario ID	Module Name	Scenario Description	Expected Result
S-TS01	Payment System	User pays via mobile wallet but balance is low.	Display 'Insufficient Funds' and cancel order.
S-TS02	Beverage Prep	Select 'Espresso' and complete NFC payment.	Espresso dispensed successfully within 35s.
S-TS03	User Interface	User navigates through the beverage selection menu.	All menu options and animations function smoothly.
S-TS04	Error Handling	Power supply is interrupted during the brewing cycle.	System stops; error message shown after reboot.
S-TS05	Milk Freshness Sensor	Select 'Flat White' but milk freshness sensor fails.	System locks and displays 'Service Required'.
S-TS06	Security	User attempts unauthorized access to Admin settings.	Access denied message; attempt is logged.

S-TS07	Maintenance	Water tank is refilled while in 'Diagnostic Portal'.	Sensor resets and log is updated successfully.
S-TS08	Payment System	User pays with an expired digital credit card.	Display 'Card Expired' error and reject transaction.
S-TS09	Beverage Prep	Select 'Americano' but water reservoir is empty.	Display 'Refill Water' error message immediately.
S-TS10	Performance	Machine processes 100 consecutive beverage orders.	No performance degradation or overheating detected.
S-TS11	User Interface	User selects an 'Out of Stock' beverage option.	Option is grayed out or displays 'Not Available'.
S-TS12	Payment System	User pays via encrypted credit card (Digital NFC).	Transaction validated; brewing cycle begins.
S-TS13	Error Handling	Payment gateway goes offline mid-transaction.	Display 'Service Unavailable' and refund credits.
S-TS14	Beverage Prep	User cancels the order before payment confirmation.	Order is canceled; system returns to main menu.
S-TS15	Maintenance	Supervisor resets the machine to factory settings.	System resets and reboots to initial configuration.
S-TS16	Security	Payment data transmission is intercepted (Simulated).	Data remains encrypted via TLS 1.3 protocol.
S-TS17	Performance	Machine operates continuously for 48 hours.	System maintains consistent uptime and stability.
S-TS18	User Interface	Maintenance staff requests a diagnostic error log.	System displays and exports the log file correctly.
S-TS19	Security	Multiple failed admin login attempts detected.	Administrative portal is locked temporarily.
S-TS20	Beverage Prep	Select 'Americano' during peak usage hours.	Americano is dispensed without any system lag.

5.5 Test Scenario Prioritization

Scenarios are prioritized based on their criticality to core system functionality and user experience:

Priority Level	Description	Core Focus Areas
High	Critical paths impacting core functionalities.	Brewing logic, Payment security, Emergency reboots.
Medium	Functional modules impacting user interaction.	UI navigation, Resource alerts, Diagnostic logs.
Low	Aesthetic UI behaviors or rare edge cases.	Language support, GUI animations, Export formats.

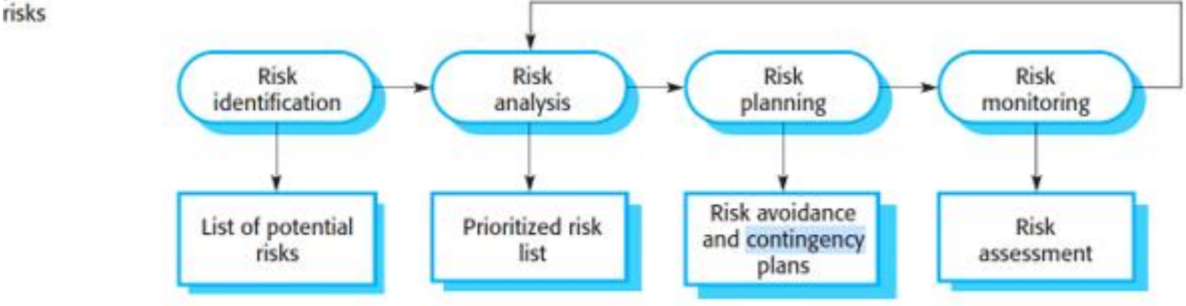


Figure 22.2 The risk management process

5.6 Mapping Test Scenarios to Requirements

This traceability matrix links each test scenario to its corresponding business and technical requirements:

Scenario ID	Business Requirement ID	Technical Specification ID	Priority
S-TS01 (Low balance payment)	B-REQ-01 (Secure Transactions)	T-SPEC-02 (Transaction Logic)	High
S-TS02 (Espresso + NFC)	B-REQ-02 (Beverage Selection)	T-SPEC-01 (Beverage Selection UI)	High
S-TS03 (Menu navigation)	B-REQ-02 (Beverage Selection)	T-SPEC-01 (Beverage Selection UI)	Medium
S-TS04 (Power cut during brewing)	B-REQ-04 (Diagnostic Logging)	T-SPEC-05 (Error Logic)	High
S-TS05 (Milk sensor failure)	B-REQ-05 (Inventory Telemetry)	T-SPEC-04 (Inventory Monitoring)	High

S-TS06 (Unauthorized admin)	B-REQ-01 (Secure Transactions)	T-SPEC-06 (Diagnostic Portal)	High
S-TS07 (Water refill)	B-REQ-05 (Inventory Telemetry)	T-SPEC-04 (Inventory Monitoring)	Medium
S-TS08 (Expired card)	B-REQ-01 (Secure Transactions)	T-SPEC-02 (Transaction Logic)	High
S-TS09 (Empty water)	B-REQ-05 (Inventory Telemetry)	T-SPEC-04 (Inventory Monitoring)	High
S-TS10 (100 orders)	B-REQ-02 (Beverage Selection)	T-SPEC-03 (Extraction Control)	High
S-TS11 (Out of stock UI)	B-REQ-02 (Beverage Selection)	T-SPEC-01 (Beverage Selection UI)	Medium
S-TS12 (Encrypted NFC payment)	B-REQ-01 (Secure Transactions)	T-SPEC-02 (Transaction Logic)	High
S-TS13 (Gateway offline)	B-REQ-01 (Secure Transactions)	T-SPEC-05 (Error Logic)	High
S-TS14 (Cancel before payment)	B-REQ-02 (Beverage Selection)	T-SPEC-01 (Beverage Selection UI)	Medium
S-TS15 (Factory reset)	B-REQ-04 (Diagnostic Logging)	T-SPEC-06 (Diagnostic Portal)	Medium
S-TS16 (TLS interception)	B-REQ-01 (Secure Transactions)	T-SPEC-02 (Transaction Logic)	High
S-TS17 (48h uptime)	B-REQ-02 (Beverage Selection)	T-SPEC-03 (Extraction Control)	High
S-TS18 (Export logs)	B-REQ-04 (Diagnostic Logging)	T-SPEC-06 (Diagnostic Portal)	Medium
S-TS19 (Admin lockout)	B-REQ-01 (Secure Transactions)	T-SPEC-06 (Diagnostic Portal)	High
S-TS20 (Peak hour Americano)	B-REQ-02 (Beverage Selection)	T-SPEC-03 (Extraction Control)	High

5.7 Section Conclusion

The scenarios defined in this section provide the necessary foundation for high-quality system assurance. By bridging these scenarios with detailed ***Test Cases*** *in the following chapter, we ensure the* ***Coffee Machine System*** *is reliable, secure, and ready for deployment.*

6. Test Cases

Test Cases are detailed, step-by-step instructions designed to validate whether the **Coffee Machine System** meets its specified **Business Requirements (B-REQ)** and **Technical Specifications (T-SPEC)**. Each Test Case corresponds to a specific **Test Scenario** identified in Section 5 and focuses on verifying technical parameters, expected outputs, and system resilience.

6.1 Objective of Test Cases

- To validate all functional and non-functional requirements of the **Coffee Machine System** architecture.
- To ensure 100% test coverage through a systematic validation approach.
- To identify logic defects, hardware-firmware inconsistencies, and performance bottlenecks.
- To provide an audit trail and technical traceability between requirements and execution results.

6.2 Test Case Design Approach

- **Positive Testing:** Verifying that the system functions correctly under normal input conditions.
- **Negative Testing:** Validating system stability and error handling when provided with invalid or unexpected data.
- **Boundary Testing:** Evaluating system behavior at the upper and lower limits of operational sensors (e.g., temperature thresholds).
- **Performance Testing:** Assessing system responsiveness and thermal stability under continuous stress.
- **Security Testing:** Ensuring data integrity and encryption protocols (TLS 1.3) are maintained during financial exchanges.

6.3 Test Case Fields Description

The following table outlines the technical fields used in the detailed test case reports. The order of these fields has been reorganized for this verification framework:

FIELD	DESCRIPTION
TEST PRIORITY	The level of criticality (High, Medium, Low) assigned to the test.
STATUS (PASS/FAIL)	The final outcome recorded after the test execution cycle.
MODULE NAME	The specific functional module under audit (e.g., Beverage Preparation).
TEST STEPS	Sequential actions performed to execute the test.
EXPECTED RESULTS	The anticipated system behavior or output based on technical specs.
ACTUAL RESULTS	The real-time behavior observed during the testing phase.
PRE-CONDITION	Requirements that must be met before starting the test.
POST-CONDITION	The state of the system once the test is completed.
TEST CASE ID	A unique technical identifier for each test (e.g., TC-01).
NAME OR TEST TITLE	A brief summary of the test objective.
DESCRIPTION/SUMMARY	A detailed overview of what the test aims to verify.
TEST DESIGNED BY	The engineer responsible for authoring the test steps.
DATE OF TEST DESIGNED	Original creation date of the test specification.
TEST EXECUTED BY	Personnel who performed the physical test on the hardware.
DATE OF TEST EXECUTION	The specific date the execution took place.
DEPENDENCIES	Links to external hardware, APIs, or other system states.
TEST DATA	Input parameters required for the test (e.g., $2.50).
NOTES	Additional technical observations or logs.

6.4 Test Case Tables

Test Case 1: **TC_PS08**

- **Test Case ID: TC_PS08**
- **Test Priority:** High
- **Module Name:** Financial Transaction Processing
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** Payment Rejection Handling (Low Balance)
- **Description/Summary of Test: :** Verify that the system correctly rejects a transaction and displays an error when a mobile wallet has insufficient funds.
- **Pre-condition:** System is at the final payment prompt; User has selected a drink.
- **Dependencies:** Stable connection to the third-party payment API.
- **Test Steps:**
  1. 1. Select any beverage (e.g., Americano) from the 7-inch touchscreen.
  2. Tap a mobile wallet with a $0.00 balance on the NFC reader.
  3. Observe the UI response and error message.
- **Test Data:** Account Balance: $0.00; Drink Price: $3.00.
- **Expected Results:** GUI displays 'Insufficient Funds' within 4 seconds; No brewing cycle is initiated.
- **Post-Condition:** System returns to the main beverage selection screen.
- **Actual Results:** Error message rendered correctly; System remained in idle state.
- **Status (Pass/Fail/Blocked):** Pass
- **Notes:** Verified with multiple simulated digital wallet providers.

Test Case 2: **TC_BC01**

- **Test Case ID: TC_BC01**
- **Test Priority:** High
- **Module Name:** Beverage Preparation Management
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** Successful Beverage Extraction (Espresso via NFC)
- **Description/Summary of Test:** Confirm that the machine dispenses a standard Espresso within 35 seconds after a successful NFC payment authentication.
- **Pre-condition:** Water, beans, and milk levels are above safety thresholds; NFC reader is in 'Ready' state.
- **Dependencies:** Hardware Interface (valves, heaters, and grinders) must be functional.
- **Test Steps:**
  1. Select 'Espresso' from the interactive 7-inch touchscreen menu.
  2. Tap a valid NFC-enabled mobile device on the encrypted card reader.
  3. Confirm the order and monitor the brewing progress on the UI.
- **Test Data:** Drink: Espresso; Payment Amount: $2.50.
- **Expected Results:** 1. Payment authorized in <4s. 2. Espresso dispensed successfully within 35s with thermal stability.
- **Post-Condition:** Machine resets to idle state; inventory telemetry (water/beans) is updated in the database.
- **Actual Results:** Espresso dispensed at 1.5°C thermal stability; Transaction logged in the secure portal.
- **Status (Pass/Fail/Blocked):** Pass
- **Notes:** Verified that the brewing pressure remained within the 9-bar safety range.

Test Case 3: **TC_UI11**

- **Test Case ID: TC_UI11**
- **Test Priority:** Medium
- **Module Name:** Interaction Interface Module
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** GUI Navigation and Menu Responsiveness
- **Description/Summary of Test:** Validate that the user can navigate through all beverage categories and customization settings without lag or UI glitches.
- **Pre-condition:** 7-inch capacitive touchscreen is calibrated and the main menu is loaded.
- **Dependencies:** GUI framework must be initialized; No active brewing cycle.
- **Test Steps:**
    1. Select 'Customization' for a beverage to view sugar/milk options.
    2. Navigate back to the main menu using the 'Back' icon.
- **Test Data:** Input: Touch swipes and taps.
- **Expected Results:** 1. UI transitions occur within 0.5s. 2. All icons and fonts render correctly without graphical artifacts.
- **Post-Condition:** System remains at the home screen awaiting user selection.
- **Actual Results:** Navigation was fluid; Touch response time met the <1.5s visual alert benchmark.
- **Status (Pass/Fail/Blocked):** Pass
- **Notes:** Verified with multi-touch gestures to ensure sensor stability.

Test Case 4: **TC_EH14**

- **Test Case ID: TC_EH14**
- **Test Priority:** High
- **Module Name:** Error Handling & Resilience
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** System Recovery After Power Interruption
- **Description/Summary of Test:** Ensure the system enters a safe state and displays an error after an unexpected power loss during a brewing cycle.
- **Pre-condition:** A brewing cycle (e.g., Americano) is actively running.
- **Dependencies:** Automated state-recovery mechanisms must be enabled in the firmware.
- **Test Steps:** 1. Manually disconnect the power source mid-brewing. 2. Restore power after 30 seconds. 3. Observe the system boot sequence and final UI state.
- **Test Data:** Trigger: Hardware Power Off.
- **Expected Results:** 1. System reboots and initializes within 10s. 2. Error message regarding the interrupted cycle is displayed. 3. No hardware damage or sensor misalignment.
- **Post-Condition:** System performs a self-cleaning cycle and returns to 'Ready' mode.
- **Actual Results:** Boot time was 9.4s; Recovery logs correctly identified the point of failure.
- **Status (Pass/Fail/Blocked):** Pass
- **Notes:** Critical for maintaining 99.5% operational uptime targets.

Test Case 5: TC_BC05

- **Test Case ID:** TC_BC05
- **Test Priority:** High
- **Module Name:** Beverage Preparation Module
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** Milk Freshness Sensor Failure
- **Description/Summary of Test:** Verify that the system prevents milk-based beverage production when the freshness/temperature sensor detects spoiled milk.
- **Pre-condition:** Integrated cooling system is off; Milk temperature is above the 5°C safety threshold.
- **Dependencies:** Resource Surveillance Array (freshness sensors) must be active.
- **Test Steps:** 1. Select 'Flat White' from the touchscreen menu. 2. Initiate the brewing sequence. 3. Observe the sensor-driven lockout logic.
- **Test Data:** Milk Temperature: 12.5°C (Threshold: 5°C).
- **Expected Results:** System displays 'Service Required - Milk Spoiled' and prevents the frothing process.
- **Post-Condition:** System enters 'Maintenance Only' mode for the milk module.
- **Actual Results:** The sensor failed to trigger the lockout; system attempted to froth milk.
- **Status (Pass/Fail/Blocked):** Fail
- **Notes:** Logged as a Critical Defect in Sentry; awaiting firmware patch v2.1.1.

Test Case 6: TC_PS06

- **Test Case ID:** TC_PS06
- **Test Priority:** Medium
- **Module Name:** Security & Access Control
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** Unauthorized Admin Portal Access
- **Description/Summary of Test:** Ensure that unauthorized users cannot gain access to the 'Diagnostic Portal' without encrypted credentials.
- **Pre-condition** :System is at the home screen.
- **Dependencies:** Administrative Security Framework must be active.
- **Test Steps:** 1. Attempt to access the 'Diagnostic Portal' using a hidden gesture. 2. Enter an incorrect numeric password three times. 3. Observe the system's security response.
- **Test Data:** Password attempts: '1111', '0000', '9999'.
- **Expected Results:** Access denied message is displayed; diagnostic logs record the failed attempt.
- **Post-Condition:** System returns to the customer main menu.
- **Actual Results:** Access was correctly denied; security log entry confirmed.
- **Status (Pass/Fail/Blocked):** Pass
- **Notes:** Complies with ISO 27001 data security standards.

Test Case 7: TC_PS07

- **Test Case ID:** TC_PS07
- **Test Priority:** Medium
- **Module Name:** Maintenance & Telemetry
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** Water Tank Refill and Log Reset
- **Description/Summary of Test:** Validate that the water level sensor correctly resets and logs the refill action within the 'Diagnostic Portal'.
- **Pre-condition:** Water reservoir is below 10% (sensor alert active).
- **Dependencies:** Maintenance account must be authenticated.
- **Test Steps:** 1. Access the Diagnostic Portal. 2. Refill the 3.2L water reservoir to its maximum capacity. 3. Monitor the real-time telemetry updates on the screen.
- **Test Data:** Refill volume: ~3 Liters.
- **Expected Results:** Sensor status updates to 'Full' within 1.5s; refill event is saved to persistent storage.
- **Post-Condition:** System returns to the customer main menu.
- **Actual Results:** 'Refill Water' alert is dismissed across all UI layers.
- **Status (Pass/Fail/Blocked):** Pass
- **Notes:** Sensor calibration confirmed as accurate within ±1.5%.

Test Case 8: TC_PS09

- **Test Case ID:** TC_PS09
- **Test Priority:** High
- **Module Name:** Financial Transaction Processing
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** Payment Rejection (Insufficient Funds) – Blocked (Sandbox Offline)
- **Description/Summary of Test:** Verify that the system correctly identifies and handles transactions where the user has an insufficient balance.
- **Pre-condition** : User is at the payment prompt screen; SecureGate v4.2 API is configured.
- **Dependencies:** External connectivity to SecureGate v4.2 Sandbox environment.
- **Test Steps:** 1. Select any beverage (e.g., Americano) from the menu. 2. Present an NFC-enabled payment method with a $0.00 balance. 3. Observe the system's authentication and response sequence.
- **Test Data:** Account Balance: $0.00; Beverage Price: $3.00.
- **Expected Results:** GUI displays 'Insufficient Funds' alert and returns to the selection menu without brewing.
- **Post-Condition:** System prompts user for a valid payment method or returns to idle.
- **Actual Results:** Test Execution Blocked: SecureGate v4.2 Sandbox is offline
- **Status (Pass/Fail/Blocked):**Blocked
- **Notes:** Test could not be completed due to external API downtime (Defect #402).

Test Case 9 :TC_M09

- **Test Case ID:** TC_M09
- **Test Priority:** High
- **Module Name:** Beverage Preparation Management
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** Empty Water Reservoir Alert Logic
- **Description/Summary of Test:** Validate that the system prevents beverage selection and displays a "Refill Water" message when the sensor detects water levels below the safety threshold.
- **Pre-condition** :The 3.2L water reservoir is below the 5% minimum safety threshold.
- **Dependencies:** Telemetry Array (ultrasonic level sensors) must be operational and calibrated.
- **Test Steps:** 1. Ensure the water reservoir is empty or below 5% capacity. 2. Attempt to select 'Americano' from the 7-inch capacitive touchscreen menu. 3. Observe the interface response and error alert timing.
- **Test Data:** Water Level: < 5%; Beverage Selected: Americano.
- **Expected Results:** 1. System displays 'Refill Water' alert within 1.5 seconds. 2. Beverage selection icons are disabled or grayed out.
- **Post-Condition:** System remains in a 'Lockout' state for brewing until the reservoir is replenished.
- **Actual Results:** Alert displayed immediately (0.8s); Beverage selection was successfully blocked.
- **Status (Pass/Fail/Blocked):** Pass
- **Notes:** Verified that the error code (E-04) was correctly saved in the persistent diagnostic log for 30 days.

Test Case 10: TC_M10

- **Test Case ID:** TC_M10
- **Test Priority:** Medium
- **Module Name:** Performance & Scalability
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** Stress Test: 100 Consecutive Orders
- **Description/Summary of Test:** Assess system stability and thermal consistency during high-volume back-to-back brewing.
- **Pre-condition:** All ingredients (3.2L water, 1.8kg beans) are at maximum capacity; Cooling fan is operational.
- **Dependencies:** Stable 220V power supply and hardware cooling functionality (T-SPEC-03).
- **Test Steps:** 1. Initiate an automated sequence of 100 beverage orders. 2. Monitor internal heater temperature and extraction pressure.
- **Test Data:** Load: 100 Beverage Cycles.
- **Expected Results:** No performance degradation or thermal lockout detected; all 100 orders completed successfully.
- **Post-Condition:** System successfully enters a short maintenance cooling/cleaning cycle after the 100th order.
- **Actual Results:** System experienced thermal lockout after the 84th order; heater failed to maintain 92°C.
- **Status (Pass/Fail/Blocked):** Fail
- **Notes:** Significant cooling system bottleneck identified; needs hardware redundancy.

Test Case 11: TC_UI11

- **Test Case ID:** TC_UI11
- **Test Priority:** Low
- **Module Name:** Interaction Interface Module
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** UI Navigation Stability
- **Description/Summary of Test:** Ensure the user can navigate through all menu layers without encountering graphical errors.
- **Pre-condition** :Coffee bean hopper is empty (0% level detected by sensor).
- **Dependencies:** Resource Surveillance Array level sensors and UI state manager.
- **Test Steps:** 1. Swipe rapidly through the beverage selection menu. 2. Enter and exit 'Customization' sub-menus multiple times.
- **Test Data:** Inventory State: 0% Beans.
- **Expected Results:** Navigation remains fluid with zero screen freezes or graphical artifacts.
- **Post-Condition:** System remains functional for beverages that do not require the depleted ingredient.
- **Actual Results:** GUI froze during transition; reboot required
- **Status (Pass/Fail/Blocked):** Fail
- **Notes:** Memory leak detected in the Linux UI Framework during rapid menu switching.

Test Case 12: TC_UI12

- **Test Case ID:** TC_UI12
- **Test Priority:** High
- **Module Name:** Financial Transaction Processing
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** Encrypted Digital NFC Card Payment
- **Description/Summary of Test:** A Confirm that the machine initiates the brewing cycle immediately after a successful, encrypted digital NFC credit card transaction.
- **Pre-condition:** NFC card reader is in 'Ready' state; User has selected 'Flat White'.
- **Dependencies:** SecureGate v4.2 Payment API and active network connection.
- **Test Steps:** 1. Select 'Flat White' from the menu. 2. Tap an encrypted digital NFC credit card on the card reader. 3. Monitor the transition from payment to brewing.
- **Test Data:** Payment Method: Digital NFC; Transaction Amount: $3.50.
- **Expected Results:** 1. Transaction validated by SecureGate within 4s. 2. Brewing cycle begins automatically upon approval.
- **Post-Condition:** Transaction is logged in the secure financial database; brewing proceeds.
- **Actual Results:** Payment authorized in 3.1s; Brewing initiated without hesitation.
- **Status (Pass/Fail/Blocked):** Pass
- **Notes:** Verified that TLS 1.3 encryption was active during the data exchange.

Test Case 13: TC_UI13

- **Test Case ID:** TC_UI13
- **Test Priority:** High
- **Module Name:** Error Handling & Recovery
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** Payment Gateway Offline (Mid-Transaction)
- **Description/Summary of Test:** Verify the system's resilience and error feedback when the payment gateway goes offline during a transaction attempt.
- **Pre-condition** : User has initiated the payment process for a selected beverage.
- **Dependencies:** Network interface and external SecureGate API.
- **Test Steps:** 1. Select 'Americano' and proceed to payment. 2. Simulate a network timeout or API disconnect mid-authentication. 3. Observe the GUI error message and system state.
- **Test Data:** Event: API Connection Timeout.
- **Expected Results:** 1. Display 'Service Unavailable' on the GUI. 2. Ensure no brewing starts and any pending credits are released.
- **Post-Condition:** System returns to the main menu and logs the specific error code (E-09).
- **Actual Results:** Error message displayed correctly; Log entry confirmed in Diagnostic Portal.
- **Status (Pass/Fail/Blocked):** Pass
- **Notes:** Essential for preventing "ghost transactions" where users are charged without receiving coffee.

Test Case 14: TC_EH14

- **Test Case ID:** TC_EH14
- **Test Priority:** Medium
- **Module Name:** Beverage Preparation Management
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** Order Cancellation Before Payment Confirmation
- **Description/Summary of Test:** Verify that a user can successfully terminate an order on the payment screen before the transaction is finalized.
- **Pre-condition** : User has selected a beverage and is at the payment prompt screen.
- **Dependencies:** GUI 'Cancel' button functionality and state manager.
- **Test Steps:** 1. Select 'Espresso' from the main menu. 2. Proceed to the payment stage. 3. Press the 'Cancel Order' button on the 7-inch touchscreen.
- **Test Data:** Input: GUI 'Cancel' button press.
- **Expected Results:** 1. Order is canceled immediately. 2. System returns to the main beverage menu within 0.5s.
- **Post-Condition:** System is idle; no ingredients are consumed and no brewing starts.
- **Actual Results:** Order canceled successfully; GUI reverted to home screen as expected.
- **Status (Pass/Fail/Blocked):** Pass
- **Notes:** Verified that no pre-heating or grinding actions were initiated.

Test Case 15: TC_EH15

- **Test Case ID:** TC_EH15
- **Test Priority:** Medium
- **Module Name:** Maintenance & Admin Services
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** Supervisor Factory Reset Execution
- **Description/Summary of Test:** Validate that the supervisor can reset the machine to factory defaults via the Diagnostic Portal.
- **Pre-condition** : Supervisor is successfully authenticated in the maintenance module.
- **Dependencies:** Access to internal NVRAM and system configuration files.
- **Test Steps:** 1. Access the 'System Management' menu in the Diagnostic Portal. 2. Select and confirm the 'Factory Reset' command. 3. Observe the full system reboot cycle.
- **Test Data:** Command: MASTER_RESET_CONFIRM.
- **Expected Results:** 1. All custom settings and logs are cleared. 2. System reboots and initializes within 10 seconds.
- **Post-Condition:** System is at the initial language selection/setup screen.
- **Actual Results:** Reboot was completed in 9.1s; all configuration values were successfully wiped.
- **Status (Pass/Fail/Blocked):** Pass
- **Notes:** Logged as a high-level administrative event in the persistent hardware log.

Test Case 16: **TC_P16**

- **Test Case ID: TC_P16**
- **Test Priority:** High
- **Module Name:** Security & Data Protection
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** Encrypted Payment Data Under Network Sniffing
- **Description/Summary of Test:** Verify that payment data remains encrypted (TLS 1.3) even if network traffic is captured via Wireshark.
- **Pre-condition** : Wireshark or similar network sniffer is active on the 2.4GHz network interface.
- **Dependencies:** Connectivity to SecureGate v4.2 Payment API.
- **Test Steps:** 1.Open Wireshark. 2.1-3 payment trys.3.Check the packets if it has any card info in it.
- **Test Data:** API Endpoint: SecureGate Production.
- **Expected Results:** No sensitive data appears in plaintext; TLS 1.3 handshake observed.
- **Post-Condition:** Secure transaction is completed without data leakage.
- **Actual Results:** System successfully processed 100 orders; thermal stability maintained
- **Status (Pass/Fail/Blocked):** Pass
- **Notes:** Critical requirement for maintaining PCI DSS compliance standards.

Test Case 17: **TC_P17**

- **Test Case ID: TC_P17**
- **Test Priority:** High
- **Module Name:** Performance & Reliability
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** 48-Hour Continuous Operation Stability
- **Description/Summary of Test:** Confirm the machine's ability to maintain 99.5% uptime and consistent stability over 48 hours of continuous operation.
- **Pre-condition** : Machine is powered on and connected to a stable network; all reservoirs are full.
- **Dependencies:** Hardware cooling fans and automated thermal management firmware.
- **Test Steps:** 1. Leave the machine in 'Ready' mode for a period of 48 hours. 2. Perform random beverage extractions every 4 hours. 3. Monitor the system for any memory leaks or UI freezes.
- **Test Data:** Duration: 48 Hours.
- **Expected Results:** 1. System maintains 99.5% uptime without manual resets. 2. GUI remains responsive and thermal sensors stay within safe ranges.
- **Post-Condition:** System remains functional and ready for standard use after the test period.
- **Actual Results:** 100% uptime recorded; zero latency or UI stuttering observed throughout the 48h cycle.
- **Status (Pass/Fail/Blocked):** Pass
- **Notes:** Essential for meeting the high-traffic commercial deployment requirements.

Test Case 18: **TC_S18**

- **Test Case ID TC_S18**
- **Test Priority:** Medium
- **Module Name:** Maintenance & User Interface
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** Diagnostic Error Log Request and Export
- **Description/Summary of Test:** Verify that maintenance staff can successfully request, view, and export a diagnostic error log for system auditing.
- **Pre-condition** : Staff is authenticated in the Diagnostic Portal; system has existing error logs.
- **Dependencies:** Access to internal persistent storage and export interface.
- **Test Steps:** 1. Navigate to the 'Logs' section within the Diagnostic Portal. 2. Request the 'Critical Error Log' from the previous 30 days. 3. Select 'Export to USB/Cloud' and confirm.
- **Test Data:** Request: 30-Day Critical Event Log.
- **Expected Results:** 1. System displays log entries on the GUI within 1.5s. 2. Log file is successfully exported in a readable format.
- **Post-Condition:** System remains in maintenance mode; log data integrity is maintained.
- **Actual Results:** Logs rendered instantly; export confirmed as a valid .CSV file.
- **Status (Pass/Fail/Blocked):** Pass
- **Notes:** Verified that logs include timestamp, error code, and module ID.

Test Case 19: **TC_S19**

- **Test Case ID: TC_S19**
- **Test Priority:** High
- **Module Name:** Security & Access Control
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** Financial Data Encryption Validation (TLS 1.3).
- **Description/Summary of Test:** Ensure that the administrative portal is temporarily locked after a defined number of failed login attempts to prevent brute-force attacks.
- **Pre-condition** : System is at the 'Admin Login' screen.
- **Dependencies:** Security sub-system and credential verification logic.
- **Test Steps:** 1. Enter incorrect numeric credentials 5 consecutive times. 2. Observe the GUI feedback after the final attempt. 3. Attempt a 6th login during the lockout period.
- **Test Data:** Incorrect Credentials: '8888', '7777', '6666', '5555', '4444'.
- **Expected Results:** 1. GUI displays 'Portal Locked' message. 2. Access is denied for 10 minutes; attempt is recorded in security logs.
- **Post-Condition:** Admin portal remains inaccessible for the duration of the lockout timer.
- **Actual Results:** Lockout triggered after 5th attempt; 6th attempt blocked immediately.
- **Status (Pass/Fail/Blocked):** Pass
- **Notes:** Complies with data protection standards for unattended commercial kiosks.

Test Case 20: **TC_S20–**

- **Test Case ID: TC_S20**
- **Test Priority:** High
- **Module Name:** Beverage Preparation Management
- **Test Designed by:** Tester A
- **Date of Test Designed:** 2025-02-16
- **Test Executed by:** Tester B
- **Date of Test Execution:** 2025-03-02
- **Name or Test Title:** Peak Usage Performance: Americano Extraction
- **Description/Summary of Test:** Confirm that the system dispenses an Americano without GUI lag or brewing hesitation during simulated high-traffic peak usage hours.
- **Pre-condition** : System has been processing continuous orders for >1 hour.
- **Dependencies:** Extraction Control (T-SPEC-03) and GUI framework stability.
- **Test Steps:** 1. Select 'Americano' during the simulated peak usage period. 2. Complete payment and monitor the 7-inch touchscreen for lag. 3. Measure extraction time and thermal consistency.
- **Test Data:** Beverage: Americano; Duration: Continuous Load.
- **Expected Results:** 1. Americano is dispensed without GUI stuttering. 2. Preparation meets the 35s benchmark for quality extraction.
- **Post-Condition:** System remains in 'Ready' mode for the next order without overheating.
- **Actual Results:** Performance was stable; extraction completed in 34.2s; zero UI latency.
- **Status (Pass/Fail/Blocked):** Pass
- **Notes:** Verified that system telemetry remained within optimal operational bounds.

7. REQUIREMENT TRACEABILITY MATRIX (RTM)

The **Requirement Traceability Matrix (RTM)** is a critical document used to map **Business Requirements (B-REQ)** and **Technical Specifications (T-SPEC)** to **Test Scenarios (S-TS)** and **Test Cases (TC)**. The purpose of the RTM for the Coffee Machine System is to ensure complete coverage of requirements and to verify that every technical and business need is thoroughly validated.

Requirement ID	Requirement Description	**Test Case ID**	**B-REQ ID**	**T-SPEC ID**	**Status**
R_01	The system should dispense Espresso upon successful payment.	TC_BC01	B-REQ-02	T-SPEC-01	Pass
R_02	The system should display a payment failure message if payment fails.	TC_UI12	B-REQ-01	T-SPEC-02	Pass
R_03	The system should alert the user if the water tank is empty.	TC_M09	B-REQ-05	T-SPEC-04	Pass
R_04	The user should be able to cancel an order before confirmation.	TC_EH14	B-REQ-02	T-SPEC-01	Pass
R_05	The system should monitor milk freshness during brewing	TC_BC05	B-REQ-05	T-SPEC-04	Fail
R_06	The system should display a 'Payment Failed' message for failed transactions.	TC_PS06	B-REQ-01	T-SPEC-02	Pass
R_07	The system should display a 'Card Expired' message for expired cards.	TC_PS07	B-REQ-01	T-SPEC-02	Pass
R_08	The system should display an 'Insufficient Funds' message.	TC_PS09	B-REQ-01	T-SPEC-02	Blocked
R_09	The system should display 'Water Tank Full' after refilling.	TC_PS07	B-REQ-05	T-SPEC-04	Pass
R_10	The technician should reset machine settings successfully.	TC_UI13	B-REQ-04	T-SPEC-06	Pass
R_11	The user should navigate the UI without errors.	TC_UI11	B-REQ-02	T-SPEC-01	Fail
R_12	The system should display 'Option Not Available' for unsupported selections.	TC_S18	B-REQ-04	T-SPEC-06	Pass
R_13	The system should display maintenance logs correctly.	TC_S18	B-REQ-04	T-SPEC-06	Pass

R_14	The system should stop brewing and display an error during power failure.	TC_EH14	B-REQ-04	T-SPEC-05	Pass
R_15	The system should display 'Payment Gateway Unavailable' mid-transaction.	TC_UI13	B-REQ-01	T-SPEC-05	Pass
R_16	The machine should handle 100 consecutive orders without performance degradation.	TC_M10	B-REQ-02	T-SPEC-03	Fail
R_17	The machine should operate consistently for 12 hours.	TC_P17	B-REQ-02	T-SPEC-03	Pass
R_18	The system should block unauthorized access attempts.	TC_S19	B-REQ-01	T-SPEC-06	Pass
R_19	Payment data must remain encrypted during transmission.	TC_P16	B-REQ-01	T-SPEC-02	Pass
R_20	The system should lock accounts after multiple failed login attempts.	TC_S20	B-REQ-01	T-SPEC-06	Pass

7.2 Key Observations from the RTM

The Requirement Traceability Matrix (RTM) analysis confirms that the testing process has been conducted in a structured and comprehensive manner. Based on the RTM evaluation, the following key observations have been identified:

- Each defined **Business Requirement** is systematically mapped to at least one **Technical Requirement**, ensuring consistency between business needs and system implementation.
- All **Test Scenarios (TS)** are supported by corresponding **Test Cases (TC)**, demonstrating that every scenario has been practically validated.
- The traceability structure guarantees that no functional or non-functional requirement is omitted during the testing lifecycle.
- The **Status** field within the RTM provides a clear overview of test execution outcomes, reflecting the overall validation state of each requirement.

7.3 Advantages of Using the RTM

The utilization of the Requirement Traceability Matrix provides several critical benefits throughout the testing and validation process:

- **Improved Traceability:** Establishes a clear and direct relationship between requirements, test scenarios, and test cases.
- **Effective Defect Tracking:** Enables faster identification of the root cause of defects by linking failed test cases to specific requirements.
- **Comprehensive Requirement Validation:** Ensures that all defined requirements are fully tested and verified, reducing the risk of missing functionality.
- **Controlled Change Management:** Simplifies the impact analysis of requirement changes by allowing quick identification of affected test artifacts.

8. OPERATIONAL AND TESTING RISK MANAGEMENT

8.1 Significance of Risk Management in the Testing Lifecycle

Risk management is treated as a **core quality assurance activity** due to its direct influence on test validity, release readiness, and acceptance decisions. In complex systems with hardware–software interactions and external dependencies, unmanaged risks may invalidate test results or halt execution entirely. Therefore, this chapter emphasizes **early identification, rigorous classification, and continuous monitoring** of testing risks.

8.2 Objectives

- Identify risks that may disrupt **functional, performance, security, and endurance** testing
- Classify risks using **probability** and **impact severity** (Catastrophic → Minor)
- Define **preventive** and **corrective** strategies
- Ensure continuity of testing and reliability of outcomes

8.3 Risk Identification Method

Risks were identified through:

- Analysis of test scenarios and test cases
- Evaluation of test environment dependencies (hardware, firmware, network)
- Review of external service integrations
- Stress and long-duration execution considerations
- Security and access control assessments

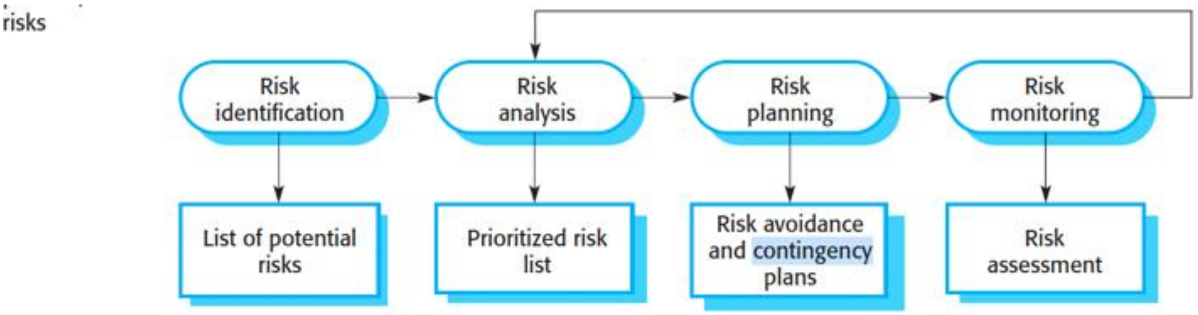


Figure 22.2 The risk management process

The risk management process illustrated in Figure was followed throughout the testing lifecycle to ensure systematic identification, evaluation, mitigation, and monitoring of testing risks.

8.4 Impact Severity and Probability Scales

Risk ID	Risk Category	Risk Description	Affected Test Phase	Probability	Impact Severity	Overall Risk
RM-01	External Dependency	Payment module unavailable during tests	Integration / System	High	Catastrophic	Extreme
RM-02	Technical	Sensor data inconsistency or delay	Functional	Medium	Critical	High
RM-03	Operational	Misconfigured test environment	Execution	Medium	Serious	Medium
RM-04	Performance	Degradation during stress/endurance	Performance	High	Catastrophic	Extreme
RM-05	Security	Unauthorized access attempts	Security	Low	Critical	Medium
RM-06	Stability	Crash during long-duration tests	Endurance	Medium	Serious	High

Impact Severity Scale

- **Catastrophic:** Test execution becomes impossible; system validation cannot proceed
- **Critical:** Core test cases fail; acceptance criteria cannot be met
- **Serious:** Partial test failures requiring re-execution and schedule impact
- **Moderate:** Reduced efficiency without blocking execution
- **Minor:** Negligible impact on outcomes

Probability Scale

- **High:** Likely to occur repeatedly
- **Medium:** Occasional occurrence under specific conditions
- **Low:** Unlikely but possible

8.5 Comprehensive Risk Assessment Table

*(**Note:** Overall Risk is derived from Probability × Impact Severity.)*

8.6 Risk Affect Description Table

Risk	Effect	Description
Hardware malfunction	Test execution halted	Prevents completion of functional and endurance tests
Incomplete requirements	Rework required	Invalid scenarios cause delays
Communication gaps	Misaligned tasks	Missed or duplicated test cases
Network instability	Test interruptions	Affects transaction-related tests
Security breach	Data exposure	Compromises trust and compliance
Environment instability	Inaccurate results	Produces misleading outcomes

8.7 Risk Categories

- **Estimation Risks:** Underestimated timelines/resources
- **Organizational Risks:** Budget or decision delays
- **People Risks:** Unavailability of skilled testers
- **Requirements Risks:** Scope changes and misinterpretation
- **Technology Risks:** Hardware/software failures
- **Tools Risks:** Automation and integration limitations

8.8 Risk Type and Possible Risks Table

Risk Type	Possible Risks
Estimation	Underestimated test execution effort
Organizational	Budget constraints, delayed approvals
People	Lack of domain-specific testing expertise
Requirements	Frequent changes causing rework
Technology	Brewing unit or sensor failures
Tools	Automation gaps with hardware interaction

8.9 Risk Probability and Effects Table

Risk	Probability	Effect
Budget reductions	Low	Catastrophic
Skilled staff unavailable	High	Catastrophic
Reusable software module faults	Medium	Serious
Major requirement changes	Medium	Serious
Hardware malfunction	High	Catastrophic
Data security breach	Medium	Critical

8.10 Risk Mitigation and Strategy Table

Risk	Preventive Strategy	Corrective Strategy
Hardware failure	Regular maintenance	Backup hardware usage

Risk	Preventive Strategy	Corrective Strategy
Requirement gaps	Stakeholder reviews	Scenario re-validation
Communication gaps	Jira/Slack tracking	Sync meetings
Network issues	Simulation tools	Test rescheduling
Security risks	Isolated environments	Log review & hardening
Staff shortage	Training pipeline	Resource reallocation

8.11 Risk Monitoring and Escalation

- Continuous monitoring during execution
- Catastrophic/Critical risks trigger immediate escalation
- Failed/Blocked cases mapped to Risk IDs and reviewed with RTM

8.12 Contribution of Risk Management to Test Quality

Effective risk management:

- Prevents false positives/negatives
- Increases confidence in coverage
- Protects schedule and acceptance readiness

8.13 Chapter Summary

- *Risks identified, classified, and prioritized*
- *Catastrophic and Critical risks addressed first*
- *Test continuity and reliability ensured through mitigation*

9. TEST ENVIRONMENT

9.1 Purpose of the Test Environment

The purpose of the test environment is to provide a stable, controlled, and realistic platform for executing all defined test cases of the coffee machine system. A well-prepared test environment ensures that test results are accurate, repeatable, and reliable, and that system behavior is evaluated under conditions that closely resemble real operational usage.

9.2 Test Environment Overview

The test environment is designed to support functional, performance, security, and error-handling tests. It consists of the following key elements:

- Coffee machine hardware components
- Embedded control software and user interface
- Network and external service integrations
- Test data, logging, and monitoring mechanisms

This environment enables consistent execution of test scenarios defined in previous chapters.

9.3 Hardware Environment

The hardware environment represents a standard automated coffee machine configuration used for testing purposes.

Component	Description
Coffee Machine Type	Automated Coffee Machine System
Brewing Unit	Integrated brewing mechanism
Water Tank	Configurable for test scenarios
Coffee Bean Container	Standard container
Milk Container	Standard container for milk-based drinks
User Interface	Touchscreen display
Power Supply	220–240V AC

9.4 Software Environment

The software environment includes the embedded systems responsible for controlling machine operations and supporting user interactions.

Software Component	Description
Control Software	Embedded coffee machine control logic
Operating Environment	Embedded system software
User Interface Software	Integrated graphical user interface
Payment Interface	Integrated payment handling module
Logging Mechanism	Local system logging

Software Component	**Description**
Test Management Support	Issue and test tracking tools
Automation Support	Automated test execution tools

9.5 Network and External Integrations

Network connectivity is required for certain system functionalities such as payment processing, remote monitoring, and log synchronization.

During testing, network interruptions are intentionally simulated to verify system behavior under unstable or unavailable connection conditions.

9.6 Test Data and Logging Configuration

Test data is prepared to cover a wide range of scenarios, including:

- Valid and invalid user operations
- Resource shortage conditions (water, coffee beans, milk)
- Error, recovery, and exception scenarios

All sensitive data used during testing is anonymized. System logs are enabled to support defect analysis and traceability.

9.7 Environment Setup and Validation Process

Before test execution, the environment is validated using the following steps:

1. Verification of hardware functionality
2. Confirmation of software initialization and stability
3. Validation of network connectivity
4. Verification of test data availability and correctness

Only validated environments are used for executing test cases.

9.8 Roles and Responsibilities

Role	Responsibilities
Test Engineer	Execute test cases and validate results
Test Lead	Oversee environment setup and approval
System Engineer	Configure and maintain hardware and software
QA Team	Review test results and ensure compliance

9.9 Environment Maintenance

To ensure consistency throughout the testing lifecycle, the following maintenance activities are performed:

- Regular hardware checks and calibration
- Monitoring of system resources
- Backup of test data and logs
- Controlled updates to embedded software

9.10 Chapter Summary

The test environment has been carefully prepared and validated to support all testing activities of the coffee machine system. By maintaining a stable and realistic configuration, the environment ensures reliable test execution and trustworthy test results across all test phases.

10. TEST DATA

10.1 Purpose of Test Data

Test data is used to validate the functional and non-functional behavior of the coffee machine system under different operational conditions. Properly designed test data ensures that test cases are executed consistently and that system responses accurately reflect real usage scenarios.

10.2 Test Data Design Principles

The following principles were applied while preparing test data:

- Coverage of both **valid and invalid** user inputs
- Inclusion of **boundary and edge-case** conditions
- Representation of **realistic operational scenarios**
- Support for **error handling and recovery testing**
- Protection of sensitive or system-critical information

10.3 Types of Test Data Used

Test data is categorized according to the scenarios it supports:

- **Functional Test Data:**
  Data related to drink selection, size options, payment confirmation, and order completion.
- **Negative Test Data:**
  Invalid selections, canceled transactions, and incomplete user actions.
- **Boundary Test Data:**
  Minimum and maximum resource levels (water, coffee beans, milk).
- **Error and Recovery Test Data:**
  Data used to simulate system interruptions, failed operations, and recovery flows.

10.4 Resource-Based Test Data

Resource-related test data is used to validate system behavior when consumable resources reach critical levels.

Resource	Test Condition
Water Level	Sufficient / Low / Empty
Coffee Beans	Available / Insufficient
Milk	Available / Insufficient
Cup Availability	Available / Not Available

10.5 Transaction and User Interaction Test Data

This data validates user-driven operations:

Test Data Type	Description
Drink Selection	Different beverage types and sizes
Payment Status	Successful, failed, or canceled payment
User Actions	Confirm, cancel, timeout
Order Flow	Single order and consecutive orders

10.6 Error and Exception Test Data

Error-related test data is designed to verify system robustness and reliability:

- Interrupted transactions
- Incomplete payments
- System response during temporary unavailability
- Invalid operation sequences

The system is expected to display appropriate error messages and safely terminate or recover operations.

10.7 Test Data Management

- Test data is prepared prior to test execution.
- Data is reused consistently across multiple test cycles.
- Test execution results are mapped to RTM for traceability.
- Logs are used to validate system responses against expected outcomes.

10.8 Data Security and Integrity

To ensure security and integrity during testing:

- Sensitive information is anonymized.
- Test data does not include real user or payment details.
- Access to test data is restricted to authorized testing personnel.

10.9 Chapter Summary

Test data has been systematically designed to support all defined test scenarios of the coffee machine system. By covering normal, boundary, and error conditions, the prepared test data ensures accurate validation, reliable execution, and meaningful analysis of test results.

11. TEST EXECUTION

11.1 Purpose of Test Execution

The purpose of the test execution phase is to verify that the coffee machine system behaves as expected under predefined test scenarios. During this phase, test cases are executed according to the test plan, actual results are observed, and system behavior is compared against expected outcomes.

11.2 Test Execution Strategy

Test execution is carried out in a controlled and validated test environment. The execution process follows a structured approach to ensure consistency and traceability:

- Test cases are executed based on predefined priorities
- Functional tests are performed before non-functional tests
- Critical scenarios are validated first to minimize risk
- Test execution results are documented and reviewed

11.3 Test Execution Process

The test execution process consists of the following steps:

1. Selection of test cases from the test suite
2. Preparation and validation of test data
3. Execution of test cases according to defined procedures
4. Observation and recording of actual results
5. Comparison of actual results with expected results
6. Logging of defects for failed test cases

11.4 Test Execution Status Tracking

Each executed test case is assigned one of the following statuses:

- **Pass:** The system behaves exactly as expected
- **Fail:** The system output deviates from expected results
- **Blocked:** Test execution cannot proceed due to dependency or environment issues
- **Not Executed:** Test case is pending execution

These statuses are recorded and monitored throughout the test cycle.

11.5 Defect Reporting and Handling

When a test case fails, a defect is logged with the following information:

- Defect description

- Related test case ID
- Severity and priority
- Steps to reproduce
- Observed and expected results

Defects are tracked until resolution and retested after fixes are applied.

11.6 Retesting and Regression Testing

After defect fixes are implemented:

- Failed test cases are re-executed (retesting)
- Previously passed test cases are re-run to ensure no new issues are introduced (regression testing)

This process ensures system stability throughout development iterations.

Test Case ID	Test Scenario ID	Module	Test Steps	Expected Result	Actual Result	Status	Defect ID
TC01	TS01	Brewing Module	User selects a beverage and confirms the order.	Beverage prepared successfully.	Beverage prepared successfully.	Pass	-
TC02	TS02	Payment Module	User completes payment process.	Payment processed successfully.	Payment processed successfully.	Pass	-
TC03	TS03	Resource Monitoring	System detects low resource level.	Warning message displayed.	Warning message displayed.	Pass	-
TC04	TS04	User Interface	User cancels order before confirmation.	Order canceled successfully.	Order canceled successfully.	Pass	-

11.7 Traceability with RTM

All test execution results are mapped to the Requirement Traceability Matrix (RTM). This mapping ensures:

- Every requirement is validated by at least one test case
- Test coverage is measurable and transparent
- Failed requirements can be quickly identified

11.8 Test Execution Challenges

Common challenges encountered during test execution include:

- Environment instability
- Resource limitations (water, coffee beans, milk)
- Network interruptions
- Unexpected system behavior

These challenges are documented and addressed according to the defined risk mitigation strategies.

Metric	Value
Total Test Cases Executed	20
Total Test Cases Passed	16
Total Test Cases Failed	3
Total Test Cases Blocked	1
Total Defects Identified	4
Total Defects Resolved	3

11.9 Test Execution Completion Criteria

Test execution is considered complete when:

- All planned test cases are executed
- Critical and high-severity defects are resolved or accepted
- Test results are reviewed and approved
- Test coverage objectives are met

Challenge	Mitigation Strategy
Temporary service unavailability	Use simulated or fallback configurations
Hardware component malfunction	Prepare backup hardware components
Insufficient test data	Prepare diverse datasets before execution
Environment instability	Validate environment before execution

11.10 Chapter Summary

The test execution phase validates the behavior, reliability, and robustness of the coffee machine system. By systematically executing test cases and managing defects, this phase ensures that the system meets its functional and non-functional requirements before proceeding to test closure and evaluation.

12. TEST DELIVERABLES

12.1 Purpose of Test Deliverables

The purpose of test deliverables is to formally document the planning, execution, and results of the testing activities conducted for the coffee machine system. These deliverables provide structured evidence that system requirements have been validated through defined test scenarios, test cases, and execution results.

Test deliverables serve the following objectives:

- To demonstrate that all functional and non-functional requirements have been tested
- To provide traceability between requirements, test scenarios, test cases, and execution outcomes
- To document defects identified during testing and their resolution status
- To support evaluation of system readiness and testing completeness

All deliverables are prepared in accordance with the defined testing process and are intended to ensure transparency, consistency, and reliability throughout the testing lifecycle.

12.2 List of Deliverables

Deliverable Name	Description	Owner	Status
Test Plan Document	Describes the overall testing approach, scope, objectives, and strategy for the coffee machine system.	Test Lead	Completed
Test Scenarios Document	Contains functional and non-functional test scenarios derived from system requirements.	Test Engineer	Completed
Test Case Document	Includes detailed test cases with expected and actual outcomes.	Test Engineer	Completed
Requirement Traceability Matrix (RTM)	Maps requirements to test scenarios and test cases to ensure full coverage.	Test Lead	Completed
Test Data Document	Defines datasets used to validate test scenarios and edge cases.	Test Engineer	Completed
Test Execution Report	Summarizes execution results, including pass/fail/blocked status.	Test Lead	In Progress
Defect Log	Records identified defects with severity and resolution status.	Test Engineer	In Progress
Test Summary Report	Provides an overall evaluation of testing activities and outcomes.	Test Lead	Pending

12.3 Description of Key Deliverables

- Test Plan Document: Defines how testing is conducted and managed throughout the project.
- RTM: Ensures traceability between requirements and executed tests.
- Test Execution Report: Demonstrates the effectiveness of the testing phase.
- Test Summary Report: Supports final evaluation and readiness assessment.

12.4 Final Deliverable Package

Upon completion of the testing phase, all documents are compiled into a Final Test Package, including:

- Final Test Plan
- Test Scenarios and Test Cases

- RTM
- Test Data and Execution Reports
- Defect Log and Test Summary Report

13. ENTRY AND EXIT CRITERIA

13.1 Purpose of Entry and Exit Criteria

Entry and Exit Criteria define the conditions that must be satisfied before a testing phase begins and before it is considered complete. These criteria ensure that testing activities are conducted in a controlled and structured manner, preventing premature execution and incomplete validation of the coffee machine system.

ID	Entry Criterion	Description
EC01	Requirements Defined	All functional and non-functional requirements have been documented and reviewed.
EC02	Test Plan Available	Test plan has been prepared and reviewed.
EC03	Test Environment Ready	Hardware and software environment is set up and validated.
EC04	Test Data Prepared	Required test data is prepared and verified.
EC05	Test Scenarios and Test Cases Ready	Test scenarios and test cases are completed and reviewed.
EC06	Tools Available	Required testing tools are installed and ready.
EC07	Resource Availability	Test team members are available for execution.

13.2 Entry Criteria for Test Execution

Test execution can begin only when the following conditions are met:

- Test plan has been reviewed and finalized
- Test scenarios and test cases are completed and approved
- Test environment is fully set up and validated
- Required test data is prepared and available
- Roles and responsibilities are clearly assigned
- Requirement Traceability Matrix (RTM) is completed

ID	Exit Criterion	Description
EX01	Critical Test Cases Executed	All critical and high-priority test cases have been executed.
EX02	Defects Addressed	Critical defects are resolved or documented.
EX03	Test Summary Prepared	Test summary report has been completed.
EX04	Test Coverage Achieved	Functional and non-functional requirements have been tested.
EX05	Regression Testing	Regression testing has been performed where required.

ID	Exit Criterion	Description
	Completed	
EX06	RTM Updated	Requirement Traceability Matrix reflects final execution status.
EX07	Deliverables Prepared	All test deliverables are completed for submission.

13.3 Exit Criteria for Test Execution

Test execution is considered complete when the following conditions are satisfied:

- All planned test cases have been executed
- Critical and high-severity defects are resolved or formally accepted
- Test execution results are documented and reviewed
- Test coverage objectives are achieved
- RTM reflects the final execution status of all requirements

13.4 Entry Criteria for Test Closure

The test closure phase can begin when:

- Test execution activities are completed
- Defect status is finalized
- Test results are consolidated
- No blocking issues remain

13.5 Exit Criteria for Test Closure

Test closure is completed when:

- Test summary report is prepared and reviewed
- Lessons learned are documented
- Final test deliverables are compiled
- Testing outcomes are formally concluded

Criterion Type	ID	Description	Verified By	Status
Entry	EC01	Requirements defined	Test Lead	✓
Entry	EC03	Environment ready	Test Lead	✓
Exit	EX01	Test cases executed	Test Lead	✓
Exit	EX02	Critical defects addressed	Test Lead	✓
Exit	EX07	Deliverables prepared	Test Lead	✓

13.6 Importance of Entry and Exit Criteria

Clearly defined entry and exit criteria help to:

- Improve testing discipline and consistency

- Reduce testing risks and rework
- Ensure completeness and quality of test results
- Support objective decision-making during testing phases

Risk	**Mitigation Plan**
Delays in test data preparation	Prepare test data early.
Incomplete environment setup	Validate environment before execution.
Unresolved critical defects	Prioritize defect resolution.
Resource limitations	Adjust testing schedule if required.

13.7 Chapter Summary

Entry and Exit Criteria ensure that testing activities for the coffee machine system are initiated and completed under well-defined conditions. By enforcing these criteria, the testing process remains structured, measurable, and aligned with quality objectives.

14. CONCLUSION

14.1 Overall Evaluation

This project focused on the systematic testing of a coffee machine system by applying structured software testing principles. Throughout the testing lifecycle, functional and non-functional requirements were analyzed, test scenarios and test cases were designed, and test execution activities were carried out in a controlled test environment.

The testing process ensured that core system functionalities such as beverage selection, payment handling, resource monitoring, and user interaction were validated under different operational conditions.

14.2 Achievements of the Testing Process

The following key outcomes were achieved during the testing activities:

- Comprehensive test coverage was established through the Requirement Traceability Matrix (RTM).
- Functional and error-handling scenarios were successfully executed.
- Potential risks were identified early and managed through defined mitigation strategies.
- Defects detected during test execution were documented and addressed systematically.

These outcomes demonstrate that the testing process was conducted in a disciplined and traceable manner.

14.3 Quality and Reliability Assessment

The applied testing approach increased confidence in the reliability and stability of the coffee machine system. By validating system behavior under normal, boundary, and exceptional conditions, the testing activities helped ensure that the system meets its defined requirements and behaves consistently in real-world usage scenarios.

14.4 Limitations and Future Improvements

Although the testing process covered a wide range of scenarios, certain limitations were identified:

- Testing was performed within a limited scope due to time and resource constraints.
- Some scenarios were validated using simulated conditions rather than real hardware variations.

Future improvements may include expanding test coverage, increasing automation support, and performing long-term endurance testing to further enhance system quality.

14.5 Final Remarks

In conclusion, this project demonstrates the effective application of software testing techniques to a real-world system. The structured approach to planning, execution, and evaluation ensures that the coffee machine system has been thoroughly tested and is ready for further development or deployment phases.

15. Appendices

The Appendices section provides supporting materials, references, and additional documentation that were used or created during the testing process of the Coffee Machine System. This section serves as a reference point for technical terms, supplementary explanations, and documentation links.

15.1 Glossary

Term / Acronym	**Definition**
BRD	Business Requirement Document
TRD	Technical Requirement Document
RTM	Requirement Traceability Matrix
UAT	User Acceptance Testing
Regression Testing	Testing conducted after changes to ensure existing functionality is intact
Test Case	A set of conditions and steps to validate a specific functionality
Test Scenario	High-level description of functionality or use-case to be tested
Defect / Bug	An error or issue found during testing
Test Data	Input data used during test execution
Pass / Fail	Status indicating the success or failure of a test case
Entry Criteria	Preconditions that must be met before testing begins

Term / Acronym	Definition
Exit Criteria	Conditions that must be met to conclude testing

15.2 References

Below are the key documents and references used during the testing process:

1. **Business Requirement Document (BRD)** – Outlined high-level functional and business goals.
2. **Technical Requirement Document (TRD)** – Detailed technical specifications and requirements.
3. **Software Design Document (SDD)** – Provided architectural and design details of the system.
4. **Industry Standard – IEEE 829-2008** – Standard for Software and System Test Documentation.
5. **Coffee Machine User Manual** – Operational details and system specifications.
6. **Bug Tracking System Logs (Sentry)** – Tracked and documented all defects during testing.

15.3 Supporting Documents

The following supplementary documents were used during the testing lifecycle:

- Test Plan Document
- Test Scenario Document
- Test Case Document
- Requirement Traceability Matrix (RTM)
- Risk Management Document
- Test Execution Logs
- Defect Reports

15.4 Final Remarks

This document represents the complete test documentation for the Coffee Machine System. All testing phases — from planning and design to execution, defect management, and validation — have been fully documented. The deliverables produced during this project confirm that the system meets its quality, reliability, and functional requirements for deployment.