**Automated Equipment Checkout System**

**Software Requirements Specification**

Victor Camacho

Version Number: 1.3

December 8, 2024

Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Revisions | Description | Author |
| 11/1/2024 | 1.0 | Initial draft | Victor Camacho |
| 11/7/2024 | 1.1 | Refined functional requirements with unique IDs, added priority use case descriptions and included initial OOAD diagrams (Use Case, Class Diagram, and VOPC Matrix). Updated Problem and Scope statements and included significant project deliverables. | Victor Camacho |
| 11/17/2024 | 1.2 | Updated Section 3.3 (Software System Attributes) with usability, reliability, performance, security, and supportability attributes. Documented architectural style in Section 4, including static and dynamic views (component and sequence diagrams). Finalized Week 3 updates. | Victor Camacho |
| 12/08/2024 | 1.3 | Incorporated functional validation test cases and updated construction/testing sections. Added detailed descriptions for testing features and risk levels. | Victor Camacho |

Contents

[1. Introduction 5](#_Toc184588203)

[1.1 Purpose 5](#_Toc184588204)

[1.2 Scope 5](#_Toc184588205)

[1.3 Definitions, Acronyms, Abbreviations 5](#_Toc184588206)

[1.4 References 5](#_Toc184588207)

[1.5 Overview 5](#_Toc184588208)

[2. Overall Description 5](#_Toc184588209)

[2.1 Product Perspective 5](#_Toc184588210)

[2.2 Product Functions 6](#_Toc184588211)

[2.3 User Characteristics 6](#_Toc184588212)

[2.4 Constraints 6](#_Toc184588213)

[2.5 Assumptions and Dependencies 6](#_Toc184588214)

[3. Specific Requirements 6](#_Toc184588215)

[3.1 External Interface Requirements 6](#_Toc184588216)

[3.1.1 User Interfaces 6](#_Toc184588217)

[3.1.2 Hardware Interfaces 7](#_Toc184588218)

[3.1.3 Software Interfaces 7](#_Toc184588219)

[3.1.4 Communications Interfaces 7](#_Toc184588220)

[3.2 Functional Requirements 7](#_Toc184588221)

[3.3 Software System Attributes 7](#_Toc184588222)

[3.3.1 Usability 7](#_Toc184588223)

[3.3.2 Reliability 7](#_Toc184588224)

[3.3.3 Performance 7](#_Toc184588225)

[3.3.4 Security 8](#_Toc184588226)

[3.3.5 Supportability 8](#_Toc184588227)

[3.4 Performance Requirements 8](#_Toc184588228)

[3.4.1 Standards 8](#_Toc184588229)

[3.4.2 Hardware Limitations 8](#_Toc184588230)

[3.5 Design Constraints 8](#_Toc184588231)

[3.5.1 Availability 8](#_Toc184588232)

[3.5.2 Security 8](#_Toc184588233)

[3.5.3 Maintainability 8](#_Toc184588234)

[3.6 Other Requirements 8](#_Toc184588235)

[4. Software Architecture Description 8](#_Toc184588236)

[4.1 Chosen Architectural Style: Layered Architecture (with Optional Tiered Approach) 8](#_Toc184588237)

[4.2 Summary 9](#_Toc184588238)

[4.3 Static View (Component Diagram) 9](#_Toc184588239)

[4.4 Dynamic View (Sequence Diagram) 10](#_Toc184588240)

[5. OOAD Diagrams 12](#_Toc184588241)

[5.1 Use Case Diagram 12](#_Toc184588242)

[5.2 Use Case Description 12](#_Toc184588243)

[5.2.1 Use Case #1: Checkout Equipment 12](#_Toc184588244)

[5.2.2 Use Case #2: Return Equipment 13](#_Toc184588245)

[5.2.3 Use Case #3: View Equipment Status 13](#_Toc184588246)

[5.2.4 Use Case #4: Report Lost Equipment 14](#_Toc184588247)

[5.2.5 Use Case #5: Generate Report 14](#_Toc184588248)

[5.3 Class Diagram 15](#_Toc184588249)

[5.4 VOPC Matrix 16](#_Toc184588250)

# Introduction

## Purpose

This document defines the Automated Equipment Checkout System requirements for GB Manufacturing. The system aims to track equipment check-in and check-out to reduce equipment loss and improve accountability within the maintenance department.

## Scope

This project aims to develop a software-based automated equipment checkout system for GB Manufacturing. The system will manage and track the check-in and check-out of tools and equipment, provide real-time tracking, and generate alerts for overdue returns. This system will be accessible by employees in the maintenance department to streamline equipment management and reduce financial losses due to lost or unreturned equipment.

## Definitions, Acronyms, Abbreviations

* **SDLC** - Software Development Life Cycle: Process for software development phases.
* **SRS** - Software Requirements Specification: Document detailing system requirements.
* **GB Manufacturing** - The company for which the system is developed.
* **UML** - Unified Modeling Language: Diagrams representing system interactions and structure.
* **OOAD** - Object-Oriented Analysis and Design: Approach for designing software using objects.
* **VOPC Matrix** - View of Participating Classes Matrix: Table linking use cases to classes.

## References

## Overview

This Software Requirements Specification (SRS) provides a detailed description of the Automated Equipment Checkout System. It includes the proposed system's purpose, scope, requirements, and design constraints. This document will guide the development and testing phases, ensuring the system meets the needs of GB Manufacturing’s maintenance department.

# Overall Description

## Product Perspective

The Automated Equipment Checkout System is designed to help GB Manufacturing’s maintenance department manage equipment checkouts and returns. By tracking each transaction and alerting employees or managers if equipment isn't returned on time, the goal is to reduce equipment loss and improve accountability.

## Product Functions

* **User Login**: Allows employees to log in securely with unique credentials.
* **Check-out and Check-in**: Tracks equipment when employees check it out and mark it as available when returned.
* **Real-time Status Tracking**: This shows the current status of each piece of equipment—whether it’s in use, available, or under maintenance.
* **Overdue Alerts**: Notifies employees and managers when equipment is overdue.
* **Usage Reports**: Provides reports for management to see how equipment is used.

## User Characteristics

* **Maintenance Employees**: The primary users will log in, check out equipment, and return it afterward.
* **Maintenance Managers**: Oversee equipment usage and receive alerts about overdue items.
* **IT Administrators**: Manage the system’s maintenance, user accounts, and data security.

## Constraints

* **Budget**: The system must be developed within the maintenance department’s budget.
* **Timeline**: The project should be completed within the current fiscal year.
* **Security**: The system must securely handle employee data and prevent unauthorized access.

## Assumptions and Dependencies

* **Assumptions**: Employees will follow the new process for checking in and out equipment.
* **Dependencies**: The system will rely on the company’s existing network to work in real-time.

# Specific Requirements

## External Interface Requirements

### User Interfaces

The system will have a web-based interface where employees can log in, view available equipment, and check items in and out.

### Hardware Interfaces

The system may need computers or tablets at equipment depots for employees to access the interface.

### Software Interfaces

The system will connect to a SQLite database that stores user details, equipment status, and transaction records.

### Communications Interfaces

The system will use secure HTTP or HTTPS protocols for data communication over the network.

## Functional Requirements

1. **REQ-001**: The system shall authenticate users with unique credentials to ensure secure access when they log in.
2. **REQ-002**: The system shall allow employees to check out equipment, updating the status to “in use” to reflect current availability.
3. **REQ-003**: The system shall allow employees to check equipment, updating the status to “available” for future use.
4. **REQ-004**: The system shall provide real-time status updates for each piece of equipment, indicating whether it is “available,” “in use,” or “overdue.”
5. **REQ-005**: To ensure accountability, the system shall automatically notify employees and managers when equipment is overdue for return.
6. **REQ-006**: The system shall generate usage reports for management, highlighting trends and patterns in equipment usage.
7. **REQ-007**: The system shall display a list of all available equipment, allowing employees to view and select items for checkout.
8. **REQ-008**: The system shall log each check-in and check-out transaction to maintain a complete record for tracking and accountability.

## Software System Attributes

### Usability

* **Importance**: The system must be easy for all employees, regardless of technical skill level. The interface should be intuitive, requiring minimal training for effective use.
* **Reasoning**: As employees interact with the system daily to check equipment in and out, a simple and user-friendly design is essential for smooth operation.

### Reliability

* **Importance**: Reliability is critical, as the system will track equipment vital to daily operations. Downtime errors could result in equipment mismanagement, impacting overall productivity.
* **Reasoning**: Ensuring that the system functions consistently, minimizes interruptions, and keeps equipment usage records accurate.

### Performance

* **Importance**: The system must operate quickly, with responses under 2 seconds for equipment check-in and check-out.
* **Reasoning**: Fast performance is necessary to avoid operations delays, as employees need to access equipment efficiently.

### Security

* **Importance**: Security is essential due to the need to protect user credentials and prevent unauthorized access.
* **Reasoning**: Protecting data, especially with user authentication, ensures that only authorized employees can access the system, which is vital for accountability and data integrity.

### Supportability

* **Importance**: The system should be easy to maintain and update as new requirements arise.
* **Reasoning**: A supportable design allows flexibility and future enhancements without overhauling the existing structure, making it adaptable as needs evolve.

Software Architecture Description

## Performance Requirements

### Standards

### Hardware Limitations

## Design Constraints

### Availability

### Security

### Maintainability

## Other Requirements

# Software Architecture Description

## Chosen Architectural Style: Layered Architecture (with Optional Tiered Approach)

Our project will adopt Layered **Architecture** as the primary design, with the potential for a Tier Architecture if additional security and scalability requirements arise. This architectural approach addresses the system’s key quality attributes identified in Section 3.3 (Software System Attributes).

**Layered Architecture (Primary):**

* **Usability:** The Layered Architecture separates the user interface (presentation layer) and the business logic, enabling a more straightforward and user-friendly interface design. This supports ease of use for employees interacting with the system.
* **Reliability:** Each layer has a dedicated responsibility, which reduces interdependencies and the likelihood of system-wide failures. This structure enhances overall system reliability by containing potential issues within specific layers.
* **Performance:** Layers can be optimized independently to enhance performance. For example, the data layer can be tuned for fast retrieval without affecting the user interface layer, supporting efficient check-in and check-out processes.
* **Supportability:** Layered architecture's modular nature simplifies updates and maintenance. Any changes or enhancements can be implemented at specific layers, minimizing the impact on the entire system.

**Optional Tiered Architecture (Secondary):**

* **Security:** The Tiered Architecture allows for physically separating layers into different network tiers if additional security is needed. For example, sensitive data in the database layer can be secured by restricting direct access to only the application logic layer.
* **Scalability:** Tiered Architecture offers flexibility to scale each tier independently, as required by usage demands. This flexibility ensures the system can handle increased data loads or user requests without compromising performance.

## Summary

Layered **Architecture** provides a flexible and reliable structure that aligns with our system’s nonfunctional requirements, such as usability, reliability, performance, and supportability. Adding the **Tiered Architecture** option provides an extra layer of security and scalability, ensuring the system meets current needs while allowing room for future growth.

## Static View (Component Diagram)

The following component diagram illustrates the layered architecture of the Automated Equipment Checkout System, highlighting the interaction between the User Interface Layer, Business Logic Layer, and Data Layer.



## Dynamic View (Sequence Diagram)

The sequence diagram below demonstrates the dynamic interaction between the Employee, User Interface Layer, Business Logic Layer, and Data Layer during the login process.



# OOAD Diagrams

## Use Case Diagram



## Use Case Description

### Use Case #1: Checkout Equipment

* **ID**: UC-001
* **Primary Actor**: Employee
* **Importance Level**: High
* **Brief Description**: Enables employees to check out equipment by verifying their credentials and updating the equipment status as "checked out."

 **Normal Flow of Events**:

1. Employee scans or enters unique employee ID.
2. System verifies employee’s ID.
3. Employee selects equipment to check out.
4. System updates equipment status to “checked out.”
5. System displays a confirmation message.

* **SubFlows**:
* If equipment is unavailable, the system displays alternative options.
* **Alternate/Exceptional Flows**:
* Invalid employee ID prompts re-entry.
* System database issue prompts a retry message.

### Use Case #2: Return Equipment

* **ID**: UC-002
* **Primary Actor**: Employee
* **Importance Level**: High
* **Brief Description**: Allows employees to return checked-out equipment, updating its status to "available."
* **Normal Flow of Events**:

1. Employee scans the equipment ID.
2. System verifies the equipment’s "checked out" status.
3. System updates the status to "available."
4. System displays confirmation.

* **SubFlows**:
* Overdue equipment alert is sent for review.
* **Alternate/Exceptional Flows**:
* Unverified equipment ID prompts re-scan.

### Use Case #3: View Equipment Status

* **ID**: UC-003
* **Primary Actor**: Supervisor
* **Importance Level**: Medium
* **Brief Description**: Allows supervisors to view all equipment's current status and location.
* **Normal Flow of Events**:

1. Supervisor logs into the system.
2. Navigate to the equipment status page.
3. System shows real-time status and location.

* **SubFlows**:
* Category filter for specific equipment types.
* **Alternate/Exceptional Flows**:
* No matching search criteria prompt message.
* Database error prompts retry later.

### Use Case #4: Report Lost Equipment

* **ID**: UC-004
* **Primary Actor**: Employee or Supervisor
* **Importance Level**: High
* **Brief Description**: A user can report equipment as lost, updating the system’s record.
* **Normal Flow of Events**:

1. User logs into the system.
2. Select “Report Lost Equipment.”
3. System flags equipment as “lost.”

* **SubFlows**:
* Allows searching by name if ID is unknown.
* **Alternate/Exceptional Flows**:
* Already flagged items prevent re-reporting.
* Connection issues prompt retry later.

### Use Case #5: Generate Report

* **ID**: UC-005
* **Primary Actor**: Supervisor
* **Importance Level**: Medium
* **Brief Description**: Enables the supervisor to generate reports on equipment usage, losses, and overdue items.
* **Normal Flow of Events**:

1. Supervisor logs into the system.
2. Navigate to the reporting section.
3. Select report type and format.
4. System generates and displays the report.

* **SubFlows**:
* Queues multiple reports sequentially.
* **Alternate/Exceptional Flows**:
* Insufficient data prompts alternative suggestions.
* Export failure suggests default format.

## Class Diagram



## VOPC Matrix

