

Software Design Document (SDD) Template

GB Manufacturing Automated Equipment Checkout System (ECS)

Version 1.0

CEIS 400 Software Engineering II
Professor Yu On Ng
DeVry University

Group G
Prepared by: Louie Venegas
03/08/2025

Team Members:
Team Member A, Team Member G, Team Member C,
Team Member B, Team Member F,
Team Member E, Team Member D

GB Manufacturing Automated Equipment Checkout System (ECS)

Contents

1.	Introduction.....	2
1.1.	Purpose.....	2
1.2.	Scope.....	2
1.3.	Definitions, Acronyms, Abbreviations.....	2
1.4.	References.....	2
1.5.	Overview.....	2
2.	Overall Description.....	2
2.1.	Product Perspective.....	2
2.2.	Product Functions.....	2
2.3.	User Characteristics.....	3
2.4.	Constraints.....	3
2.5.	Assumptions and Dependencies.....	3
3.	Specific Requirement.....	3
3.1.	External Interface Requirements.....	3
3.2.	Functional Requirements.....	3
3.3.	Performance Requirements.....	4
3.4.	Design Constraints.....	4
3.5.	Other Requirements.....	4
4.	Appendices.....	4
4.1.	System Architecture Diagrams.....	4
4.2.	Use case diagrams.....	4
4.3.	Database schema.....	4

1.1.

1. Introduction

1.1. Purpose

- 1.1.1 This document outlines the software requirements for the Equipment Checkout System (ECS). This system will improve the efficiency and accuracy of the equipment check-in/check-out processes for GB Manufacturing's maintenance department, minimizing losses and enhancing accountability.

1.2. Scope

- 1.2.1 The ECS will streamline equipment management by automating check-ins and check-outs, tracking inventory, and enforcing employee accountability. The system will integrate with barcode scanners and a MySQL database while being developed in Java with Apache Ant in NetBeans.

1.3. Definitions, Acronyms, Abbreviations

- 1.3.1 ECS: Equipment Checkout System
- 1.3.2 FR: Functional Requirement
- 1.3.3 NFR: Non-Functional Requirement
- 1.3.4 SQL: Structured Query Language

1.4. References

- 1.4.1 IEEE 830 Software Requirements Specification Standard.
- 1.4.2 GB Manufacturing Internal Equipment Tracking Policies.

1.5. Overview

- 1.5.1 This document details the functional and non-functional requirements of ECS, external interfaces, constraints, and dependencies.

2. Overall Description

2.1. Product Perspective

- 2.1.1 ECS is a standalone system designed to replace the existing manual tracking method. It will integrate with GB Manufacturing's internal systems for reporting and equipment tracking.

2.2. Product Functions

- 2.2.1 Automated Equipment Check-In/Check-Out
- 2.2.2 Real-Time Inventory Tracking
- 2.2.3 Employee-Based Access Control

2.2.4 Reporting and Analytics for Equipment Usage

2.2.5 Barcode Scanner Integration

2.3. User Characteristics

2.3.1 Maintenance Employees: Use the system to borrow and return equipment.

2.3.2 Equipment Depot Staff: Manage inventory and ensure equipment availability.

2.3.3 Supervisors: Monitor tool usage and ensure compliance.

2.3.3.1. IT Team: Maintain and support the system.

2.3.3.2. HR Staff:

2.4. Constraints

2.4.1 The system must be accessible only within GB Manufacturing's internal network.

2.4.2 The system must comply with company security policies.

2.5. Assumptions and Dependencies

2.5.1 Users will be trained on how to use the barcode scanners.

2.5.2 The system will rely on an existing employee database for authentication

3. Specific Requirement

3.1. External Interface Requirements

3.1.1 User Interfaces

3.1.1.1. The system shall provide a web-based dashboard for tracking check-ins and check-outs.

3.1.2 Hardware Interfaces

3.1.2.1. The system shall integrate with barcode scanners to process equipment transactions.

3.1.3 Software Interfaces

3.1.3.1. The system shall connect with a MySQL database for storing transaction records.

3.1.4 Communications Interfaces

3.1.4.1. The system shall support email notifications for overdue equipment.

3.2. Functional Requirements

3.2.1 Equipment Management Subsystem

3.2.1.1. FR1: The system shall allow employees to check out equipment using barcode scanners.

3.2.1.2. FR2: The system shall maintain a record of all equipment transactions.

3.2.1.3. FR3: The system shall notify supervisors of overdue equipment.

3.2.2 Access Control Subsystem

3.2.2.1. FR4: The system shall restrict access to specific tools based on employee roles.

3.2.2.2. FR5: The system shall authenticate users before allowing equipment checkouts.

3.2.3 Notifications and Alerts Subsystem

3.2.3.1. FR6: The system shall send automated email notifications for overdue equipment.

3.2.3.2. FR7: The system shall provide real-time alerts when stock levels are low.

3.2.4 Reporting

3.2.4.1. FR5: The system shall generate reports on equipment usage and losses.

3.3. Performance Requirements

3.3.1 Standards

3.3.1.1. The system shall comply with IEEE 830 Software Requirements Specification standards.

3.3.1.2. The database shall follow ANSI SQL standards for compatibility

3.3.2 Hardware Limitations

3.3.2.1. The system shall operate on standard desktop workstations used within GB Manufacturing.

3.3.2.2. The barcode scanners must be USB or Bluetooth-compatible with the system.

3.4. Design Constraints

3.4.1.1. Availability

3.4.1.2. The system shall be available 99.9% of the time.

3.4.2 Security

3.4.2.1. The system shall encrypt sensitive employee data.

3.4.3 Maintainability

3.4.3.1. The system shall support regular software updates without downtime.

3.5. Other Requirements

3.5.1.1. The system shall log all equipment transactions for auditing purposes.

4. Appendices

4.1. System architecture diagrams

4.2. Use case diagrams

4.3. Database schema

1.2.

Project Description

The Equipment Checkout System (ECS) is a digital solution designed to streamline the process of checking in and checking out equipment for maintenance employees at GB Manufacturing. The system will provide an intuitive user interface for employees to request and return equipment efficiently, reducing processing time and minimizing losses. ECS will also enforce accountability by tracking each employee's equipment transactions and restricting access to specific tools based on skill classifications. The system will generate real-time reports on equipment usage, availability, and loss trends for better decision-making and inventory management.

From a technical standpoint, ECS will be developed using Java and Apache Ant in the NetBeans IDE for both frontend and backend development. The system will be built on a MySQL database to efficiently store and manage equipment records. ECS will also integrate barcode scanning technology to automate check-ins and check-outs, reducing manual input errors. The development team will leverage version control tools for tracking code changes and ensuring collaboration, along with automated testing strategies to maintain code quality and system reliability. The system will be deployed securely to provide stable and scalable access across multiple GB Manufacturing locations.

Project Objectives

- Reduce equipment checkout processing time to under 2 minutes per transaction.
- Decrease annual equipment losses by at least 50% through automated tracking and accountability measures.
- Implement real-time equipment availability status to reduce delays in locating necessary tools.
- Ensure only authorized employees with proper skill classifications can check out specific equipment.
- Generate automated reports on equipment usage, losses, and maintenance needs to aid decision-making.
- Improve employee accountability by providing periodic statements of checked-out equipment.
- Enhance security by integrating role-based access control for equipment transactions.
- Minimize excess inventory and carrying costs by optimizing resource allocation.
- Provide a user-friendly interface to simplify the check-in/check-out process for employees and depot staff.

Efficient Equipment Checkout System Test Plan

1. Introduction

This test plan outlines the testing approach for the GB Manufacturing Automated Equipment Checkout System (ECS). The primary objective is to improve efficiency by reducing processing time, ensuring data accuracy, enforcing equipment skill classification requirements, and automating notifications and reporting.

2. Test Items

The following items will be tested in the ECS:

- Equipment Checkout Processing Time
- Employee Termination Process
- Returned Equipment Matching
- Skill Classification Enforcement
- Equipment Order Tracking
- Record Accessibility
- Report Generation
- Periodic Transaction Statements
- Identifying Excessive Equipment Losses
- Problem Employee Follow-Up

3. Features to be Tested

- Automated data entry using manual input fields instead of barcode scanning.
- Real-time validation of skill classifications for equipment checkout.
- Automated notification system for equipment orders and overdue items.
- System-generated exception reports for excessive equipment losses.
- Quick retrieval of employee and equipment records.
- Integration with a MySQL database for tracking and reporting.

4. Features Not to Be Tested

- User manual and help documentation
- Mobile or platform-specific functionality outside of the core system.

5. Approach

Testing will be performed in the following stages:

1. Unit Testing: Individual components such as login, equipment search, and checkout processes will be tested.
2. Integration Testing: Testing the interaction between modules such as searching for equipment, checking out equipment, and updating records.
3. System Testing: The entire Equipment Checkout System will be tested for expected behavior.
4. Acceptance Testing: Ensuring the system meets the requirements and end-user needs.

The test cases will be designed based on the system requirements. Both positive and negative test cases will be written to ensure the system handles all possible scenarios.

6. Pass/Fail Criteria

- A test will be considered **passed** if the system behaves as expected under the specified conditions and satisfies the acceptance criteria.
- A test will be considered **failed** if the system deviates from the expected behavior, crashes, or produces incorrect outputs.

Each test item has the following pass/fail criteria:

1. Equipment Checkout Processing Time
 - o Pass: Processing time is less than 2 minutes.
 - o Fail: Processing time exceeds 2 minutes.
2. Employee Termination Process
 - o Pass: Termination process completes in under 2 minutes.
 - o Fail: Termination process exceeds 2 minutes.
3. Returned Equipment Matching
 - o Pass: Equipment is matched with the correct employee within 30 seconds.
 - o Fail: Matching exceeds 30 seconds or fails.
4. Skill Classification Enforcement
 - o Pass: System prevents unauthorized checkout.
 - o Fail: Unauthorized employees can check out restricted equipment.
5. Equipment Order Tracking
 - o Pass: System correctly notifies employees upon order arrival.
 - o Fail: Notifications are delayed or do not occur.
6. Record Accessibility
 - o Pass: Records can be accessed within 5 seconds.
 - o Fail: Retrieval exceeds 5 seconds or fails.
7. Report Generation
 - o Pass: Reports generate within 5 seconds and display accurate data.
 - o Fail: Report generation takes too long or is incorrect.
8. Periodic Transaction Statements
 - o Pass: Employees receive accurate and timely statements.
 - o Fail: Statements are missing, inaccurate, or delayed.
9. Identifying Excessive Equipment Losses
 - o Pass: System flags employees with excessive losses.
 - o Fail: Reports fail to generate or miss problem employees.
10. Problem Employee Follow-Up
 - o Pass: System tracks problem employees and logs follow-up actions.
 - o Fail: Issues are not logged or cannot be tracked.

7. Suspension Criteria

Testing will be suspended if:

- Major bugs or defects are found that prevent further testing.
- Critical system components are unavailable for testing.

8. Test Deliverables

- Test plan document

- Test cases document
- Test execution results
- Test summary report

9. Testing Schedule

Phase	Start Date	End Date
Test Planning		
Test Case Development		
Test Execution (Unit)		
Test Execution (Integration)		
System Testing		
Acceptance Testing		

10. Resource Requirements

- **Personnel:**
 - Project Manager
 - Software Developers
 - End users for input and direction
- **Software:**
 - Development programs
 - Testing tools
 - Bug tracking system
- **Hardware:**
 - Server to host the ECS application
 - Workstations for testers and developers

11. Environmental Needs

- A test environment that closely mirrors the production environment.
- A stable network connection to access and test the application.
- Access to the ECS database to simulate real-life scenarios for checkout and return.

12. Risks and Contingencies

- Risk: System Load
 - Contingency: Ability to address performance issues of transaction volume and speed.
- Risk: Unexpected bugs found during testing may delay the release.
 - Contingency: Allocate extra time in the schedule for bug fixes and re-testing.

13. Test Case Example

Test Case ID	Test Case Name	Test Description	Expected Result	Pass/Fail
TC001	Login	Test valid login for a	User is logged in and	Pass

Software Design Document (SDD) Template

	Functionality	customer user.	redirected to the homepage.	
TC002	Checkout Process	Test equipment checkout with valid user and available equipment.	Equipment is successfully checked out and updated in the database.	Pass
TC003	Return Process	Test the return process of checked-out equipment.	Equipment status is updated to "returned" and available for checkout.	Pass
TC004	Overdue Equipment Notification	Test overdue notification after the equipment return deadline.	User receives an overdue notification if equipment is not returned on time.	Pass

14. Approval

This test plan is approved by:

Role	Name	Signature	Date
Project Manager	Louie Venegas		
Developer(s)	Team Member E	ML	03/15/2025

This test plan outlines the necessary components for testing the Equipment Checkout System. It includes the features to be tested, the approach to testing, the criteria for success, and all logistics involved in testing.

Justification for Three-Tier Layered Architecture

The Equipment Checkout System (ECS) will adopt a three-tier layered architecture to ensure a structured, maintainable design that meets performance, security, and scalability needs. This approach clearly separates the user interface, business logic, and data storage into distinct tiers for clarity and easier ongoing development.

Additionally, the system will leverage NetBeans (for development), Java-based authentication (for authentication and authorization services), MySQL (for database storage), and a Windows Server platform (for hosting and deployment).

Tier Breakdown

1. Presentation Tier

- **Purpose:** Delivers the user interface for employees, supervisors, and IT staff. Emphasis is on a clean layout, straightforward data entry, and responsive feedback.
- **Implementation:** Built using Java in NetBeans, enabling a consistent and portable user experience that can run on a Windows Server environment or other compatible OS as needed.

2. Business Logic Tier

- **Purpose:** Centralizes the rules and processes for equipment checkout, return, notifications, and authentication. Ensures consistent policy enforcement across the system.
- **Implementation:** Developed in NetBeans and deployed on Windows Server, with Java-based authentication to manage user roles, permissions, and access control.

3. Data Tier

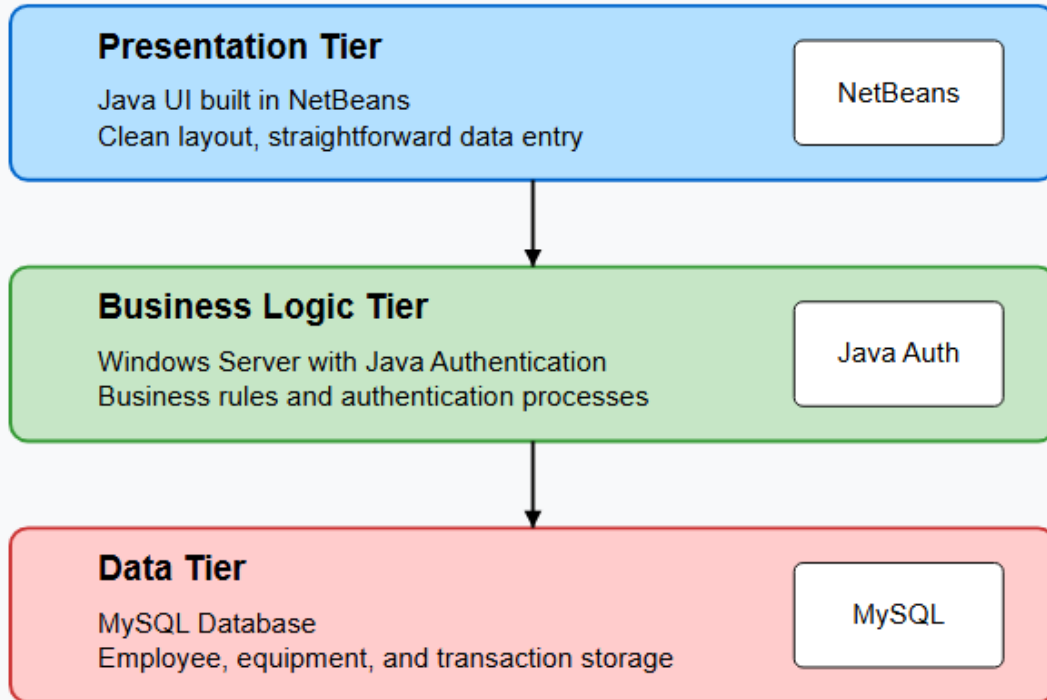
- **Purpose:** Stores all employee, equipment, and transaction records in a secure, organized database. Indexing and backup procedures help maintain reliability and ensure fast retrieval.
- **Implementation:** Uses MySQL, taking advantage of its stability, indexing features, and compatibility with Windows Server deployments.

Key Advantages

- **Straightforward Scalability:** Each tier (UI, business logic, data) can be deployed or expanded independently, allowing the system to handle increasing load.
- **Clear Separation of Concerns:** Rapid updates to the user interface, business rules, or database can be implemented without negatively impacting other tiers.
- **Enhanced Security:** A single point of responsibility for access control within the business logic tier reduces risk and simplifies authentication workflows. The Java-based authentication system controls user access based on employee roles and skill classifications.
- **Unified Development Environment:** Using NetBeans for both presentation and business logic tiers creates consistency in the development process and simplifies integration between these layers.

By adopting this three-tier layered architecture with NetBeans, Java authentication, MySQL, and Windows Server, the ECS remains flexible enough to accommodate new features, reliable under increasing transaction loads, and secure in safeguarding sensitive data while enforcing the skill-based checkout requirements.

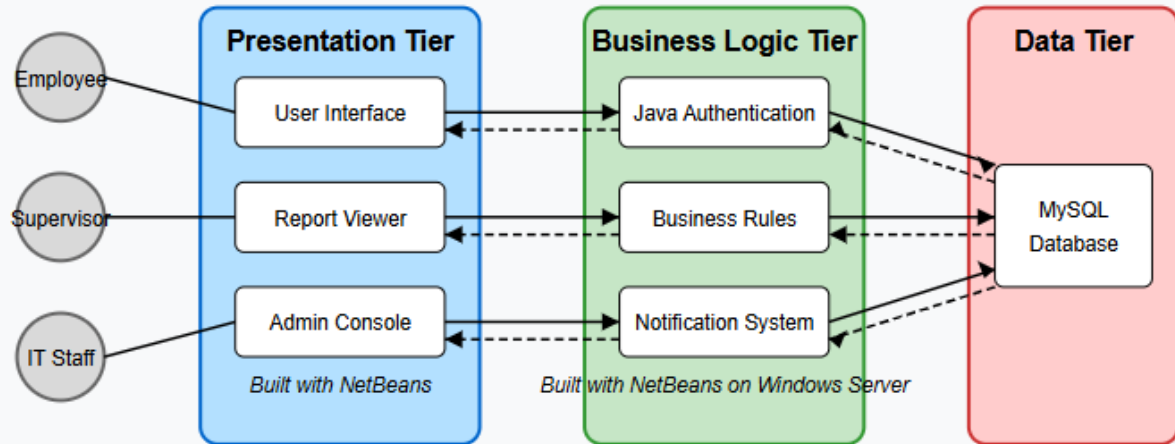
Three-Tier Architecture: Equipment Checkout System



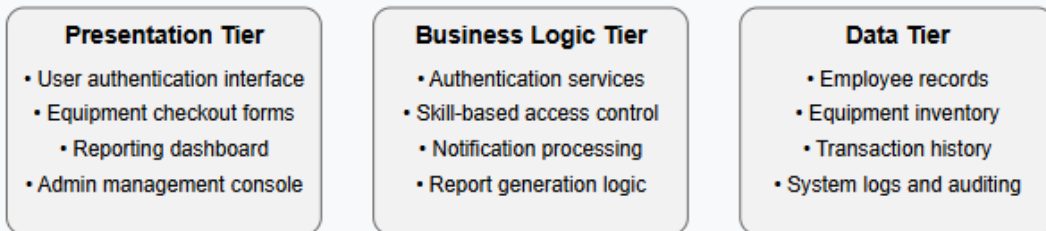
Key Advantages

- Straightforward Scalability
- Clear Separation of Concerns
- Enhanced Security

Equipment Checkout System: Data Flow Architecture



Key System Functions



Use Case Diagram Syntax

AN ACTOR:

- Is a person or system that derives benefit from and is external to the system
- Is labeled with its role
- Can be associated with other actors using a specialization/superclass association, denoted by an arrow with a hollow arrowhead
- Is placed outside the system boundary



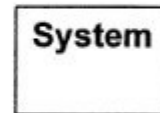
A USE CASE:

- Represents a major piece of system functionality
- Can extend another use case
- Can include another use case
- Is placed inside the system boundary
- Is labeled with a descriptive verb-noun phrase



A SYSTEM BOUNDARY:

- Includes the name of the system inside or on top
- Represents the scope of the system



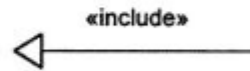
AN ASSOCIATION RELATIONSHIP:

- Links an actor with the use case(s) with which it interacts



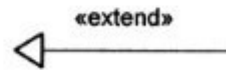
AN INCLUDE RELATIONSHIP:

- Represents the inclusion of the functionality of one use case within another
- The arrow is drawn from the base use case to the used use case



AN EXTEND RELATIONSHIP:

- Represents the extension of the use case to include optional behavior
- The arrow is drawn from the extension use case to the base use case



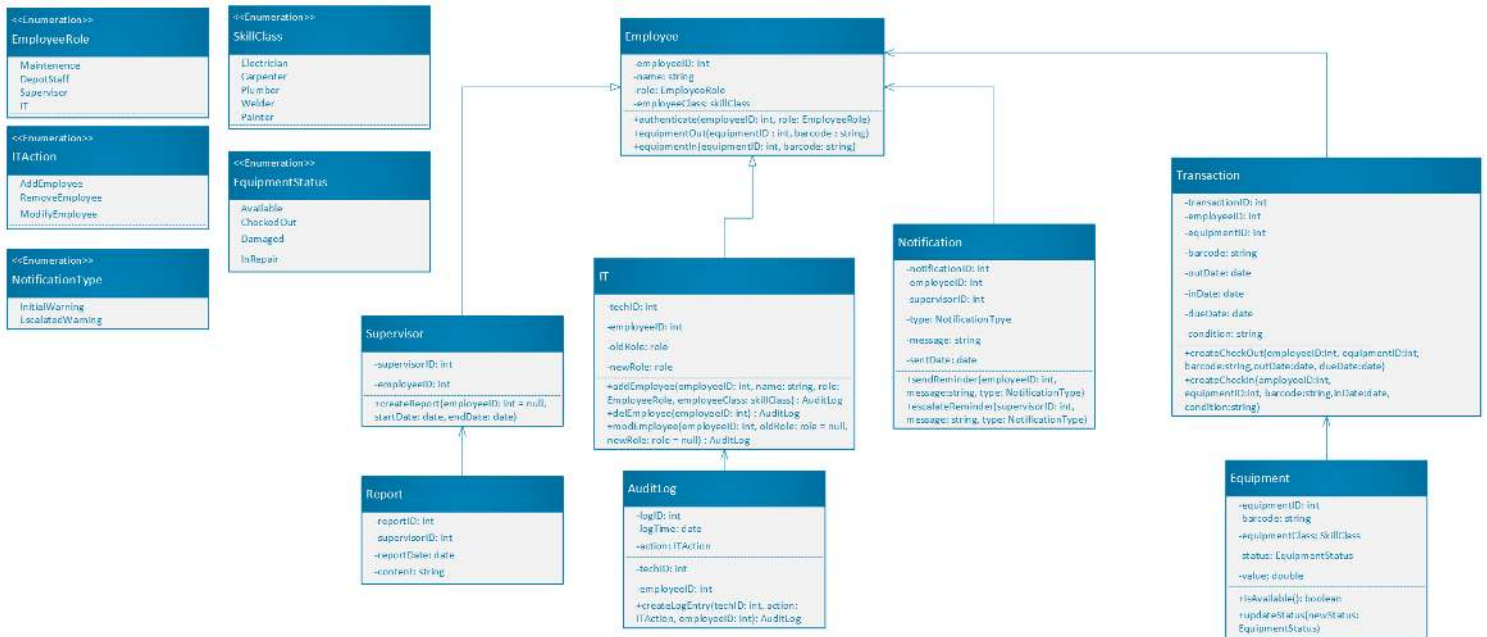
A GENERALIZATION RELATIONSHIP:

- Represents a specialized use case to a more generalized one
- The arrow is drawn from the specialized use case to the base use case



Software Design Document (SDD) Template

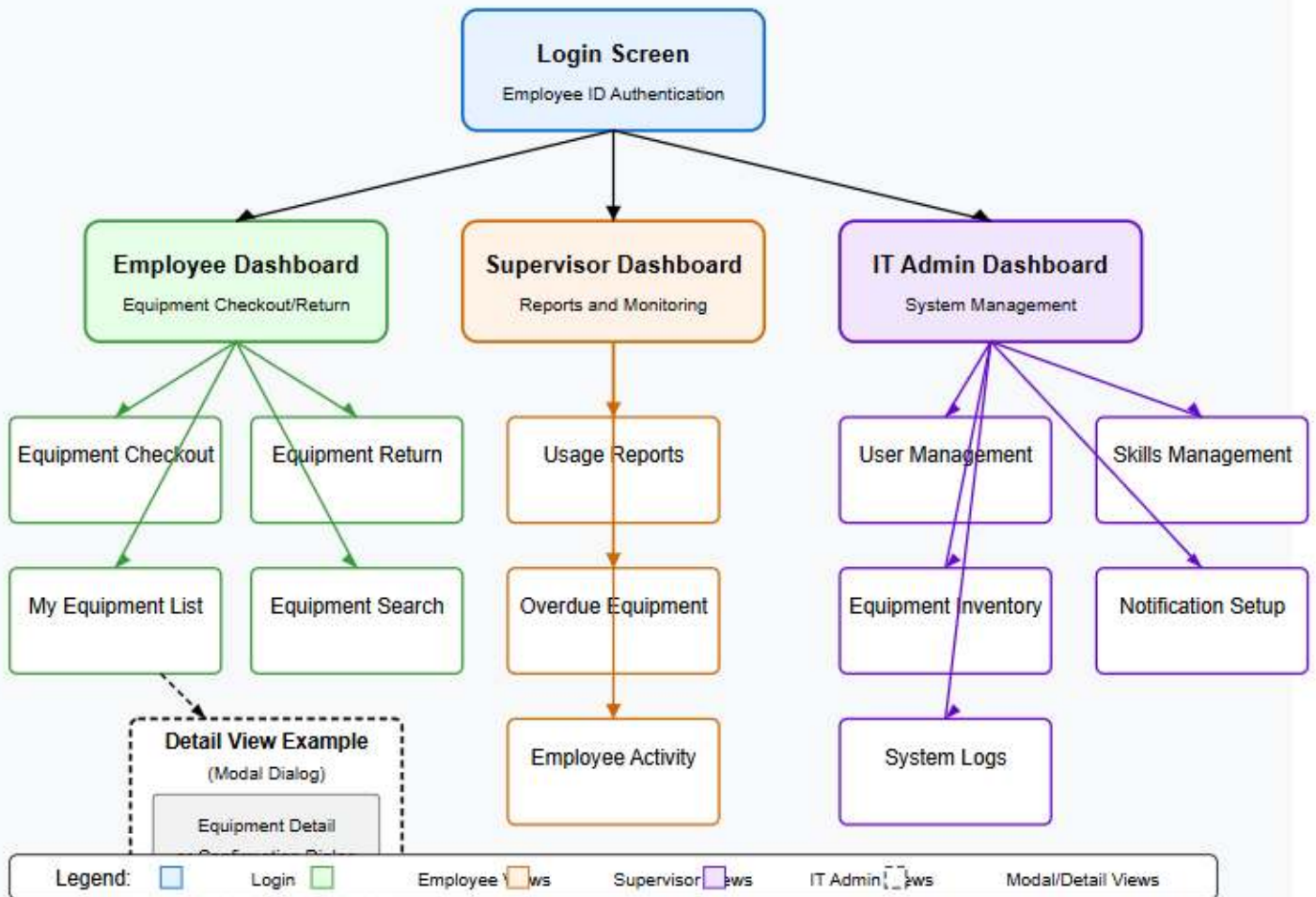
Class Diagram



View Of Participating Classes Matrix

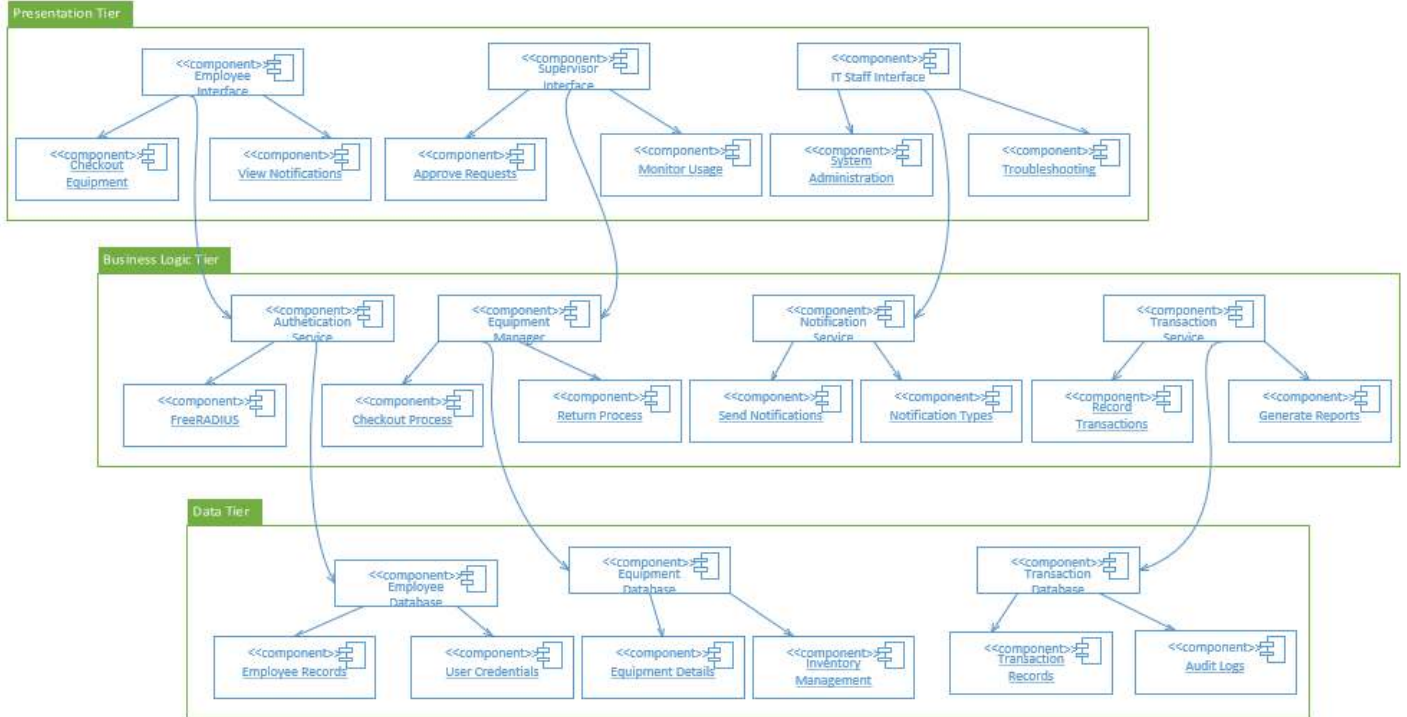
USE CASES / CLASSES	Equipment Checkout	Equipment Check-in	Update Equipment Inventory	Add Employee	Delete Employee	Modify Employee	Generate Reports	Send Notification
Employee	X	X	X				X	X
IT				X	X	X		
Supervisor							X	
Auditing				X	X	X		
Report							X	
Transaction	X	X	X					
Equipment	X	X	X					
Notification								X

Equipment Checkout System: Dynamic View Flow

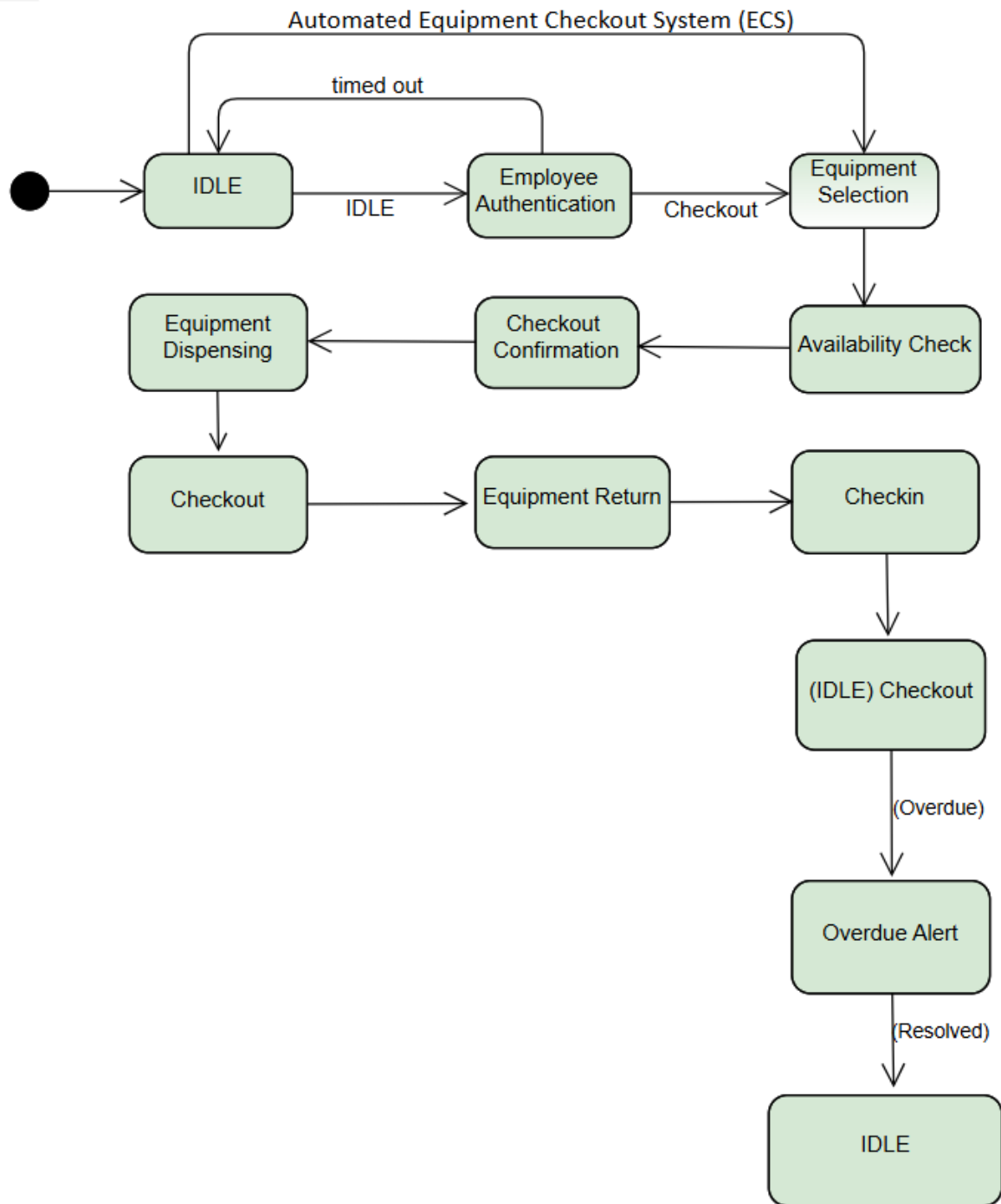


Static View

Software Design Document (SDD) Template



Software Design Document (SDD) Template



Use Case Description

Use Case Name: Check Out Equipment	ID: UC1	Importance Level: High
Primary Actor: Maintenance Employee	Use Case Type: User-Initiated	

Software Design Document (SDD) Template

Stakeholders and Interests: Maintenance Employees, Equipment Depot Staff	
Brief Description: This use case allows employees to check out equipment from the Equipment Checkout System (ECS).	
Trigger: Employee initiates the checkout process.	
Type: Manual	
Relationships:	
Include: Authenticate User	
Extend: Check Equipment Availability	
Normal Flow of Events:	
-Employee types their ID to authenticate. - Employee types the barcode of the equipment. - The system verifies availability and assigns the item to the employee. - The system logs the transaction and updates the inventory. - System notifies the employee of the due date.	
SubFlows:	
S-1: Authenticate User: The system verifies the employee ID and permission level.	
S-2: Verify Equipment Availability: The system checks if the equipment is available for checkout.	
Alternate/Exceptional Flows:	
- If the employee is not authorized, the system denies the checkout. - If the equipment is not available, the system notifies the employee and suggests alternatives.	

Use Case Name: Check In Equipment	ID: UC2	Importance Level: High
Primary Actor: Maintenance Employee	Use Case Type: User-Initiated	
Stakeholders and Interests: Maintenance Employees, Equipment Depot Staff		
Brief Description: This use case allows employees to return checked-out equipment to the Equipment Checkout System (ECS).		
Trigger: Employee initiates the check-in process.		
Type: Manual		
Relationships:		
Include: Authenticate User		
Extend: Generate Condition Report		
Normal Flow of Events:		
- Employee types ID for authentication. - Employee types the barcode of returning equipment. - System verifies return validity and updates inventory. - If the item is damaged, the system prompts for a condition report.		

Software Design Document (SDD) Template

SubFlows:

- S-1: Authenticate User: The system verifies the employee ID and permission level.
- S-2: Log Equipment Condition: If the item is damaged, the system records it for maintenance tracking.

Alternate/Exceptional Flows:

- If the employee returns an unregistered item, the system flags it for manual review.
- If the equipment is returned late, the system logs a late return report.

Use Case Name: Generate Equipment Usage Report	ID: UC3	Importance Level: High
Primary Actor: Supervisor	Use Case Type: System-Initiated	
Stakeholders and Interests: Supervisors, IT Team		
Brief Description: This use case allows supervisors to generate reports on equipment usage.		
Trigger: Supervisor requests a report		
Type: Manual		
Relationships:		
Include: Check Out Equipment, Check In Equipment		
Normal Flow of Events:		
- Supervisor logs into the system. - The supervisor selects a reporting period. - System retrieves equipment transaction logs. - The system generates a detailed usage report. - The supervisor exports or prints the report.		
SubFlows:		
S-1: Retrieve Transaction Logs: The system gathers checkout and check-in records within the selected time frame.		
S-2: Format Report Data: The system formats the report into a readable document.		
Alternate/Exceptional Flows:		
- If no data is available for the selected period, the system notifies the supervisor.		

Use Case Name: Notify Overdue Equipment	ID: UC4	Importance Level: High
Primary Actor: ECS (Automated Process)	Use Case Type: System-Initiated	
Stakeholders and Interests: Supervisors, Maintenance Employees		

Software Design Document (SDD) Template

Brief Description: This use case notifies employees and supervisors of overdue equipment.

Trigger: System detects overdue equipment.

Type: Automated

Relationships:

Include: Check Out Equipment

Extend: Escalate to Supervisor

Normal Flow of Events:

- The system scans the database for overdue checkouts.
- The system sends email reminders to employees.
- If overdue beyond the threshold, the system escalates to supervisors.

SubFlows:

S-1: Send Reminder: The system sends a notification to the employee about overdue equipment.

S-2: Escalate to Supervisor: If the due date passes a threshold, the system alerts the supervisor for further action.

Alternate/Exceptional Flows:

- If an employee no longer works for the company, the system alerts HR for retrieval.

Use Case Name: Manage User Access and Permissions	ID: UC5	Importance Level: High
Primary Actor: IT Team	Use Case Type: Admin-Initiated	
Stakeholders and Interests: IT Team, Supervisors		
Brief Description: This use case allows IT staff to manage user access and permissions within ECS.		
Trigger: IT team modifies user permissions.		
Type: Manual		

Software Design Document (SDD) Template

Relationships:

Include: Authenticate User

Normal Flow of Events:

- IT team logs into the system.
- The IT team adds/removes users or modifies roles.
- System updates permissions accordingly.
- Changes are logged for auditing.

SubFlows:

- S-1: Authenticate IT User: The system verifies IT credentials before allowing modifications.
- S-2: Log Changes: Every modification is recorded for auditing purposes.

Alternate/Exceptional Flows:

- If the IT team accidentally revokes access, they must manually restore it.