Equipment Checkout System (ECS) — Software Design Description (SDD)

Prepared by: Group 9  
Date: 2025-08-10  
Version: v1.0

Purpose: Define the component-level design for ECS and provide a complete blueprint for construction and testing.

# Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Change | Author |
| 1.0 | 2025-08-10 | Complete SDD with updated RTM and diagrams. | Group 9 |

# 1. Introduction

## 1.1 Purpose

Provide a complete and organized blueprint for implementing ECS.

## 1.2 Scope

Barcode-based tool checkout/check-in with custody tracking, condition capture, and audit history.

## 1.3 References

- Team 9 IEEE-830 SRS  
- Team 9 Software Architecture Description  
- IEEE Std 1016-2009

## 1.4 Definitions & Acronyms

ECS, UI, API, DAO.

# 2. System Overview

Users log in, scan a tool, perform checkout or check-in. System validates state, updates custodian, logs a transaction, and provides feedback.

# 3. System Architecture

## 3.1 Architectural Style & Rationale

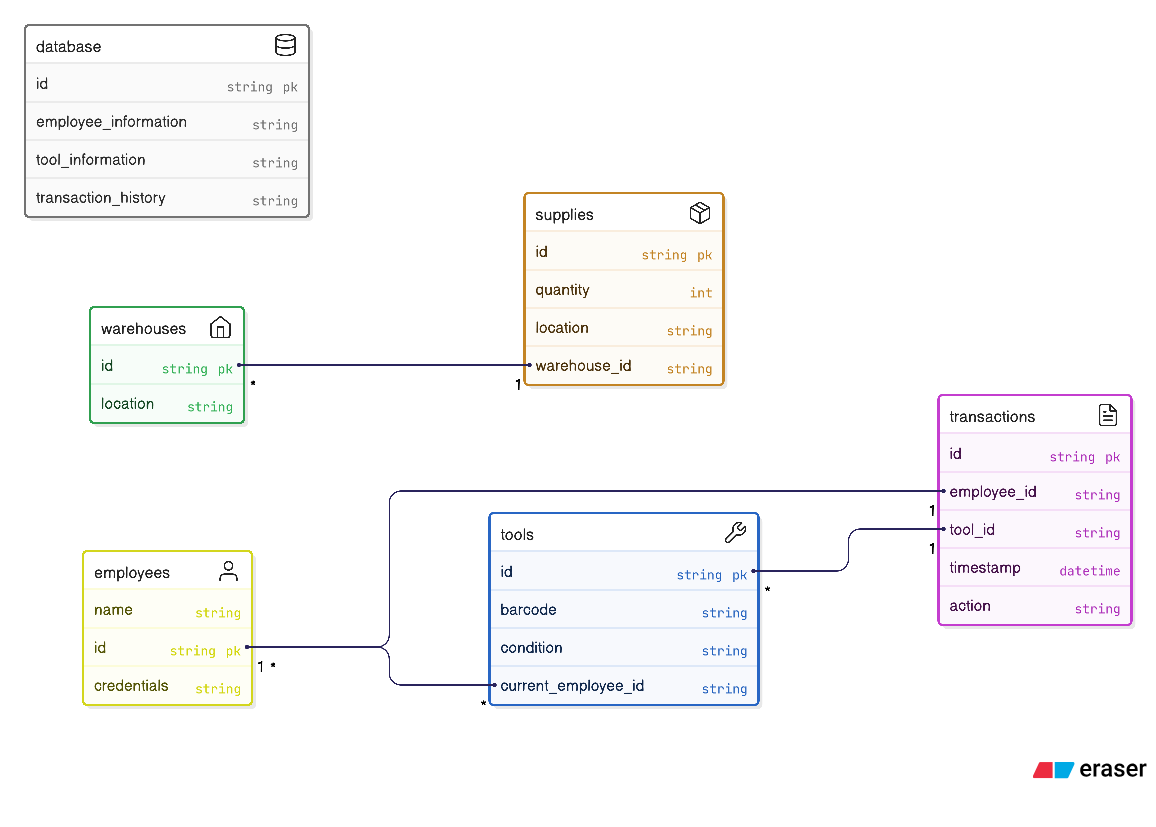
Repository-centered architecture with layered services (UI → Services → Domain → Repositories → DB) for clarity, testability, and incremental build.

## 3.2 Logical View

UI screens (Login, Scan, My Tools, History) call services (AuthService, ToolService, TransactionService, ReportingService) that operate on domain entities (Employee, Tool, Transaction, Warehouse, Supply) via Repositories/DAOs.

## 3.3 Data View (ER/Schema)

Figure: ERD / Data Model.



## 3.4 Deployment View

Single database with client app(s); future API server optional.

# 4. Data Dictionary

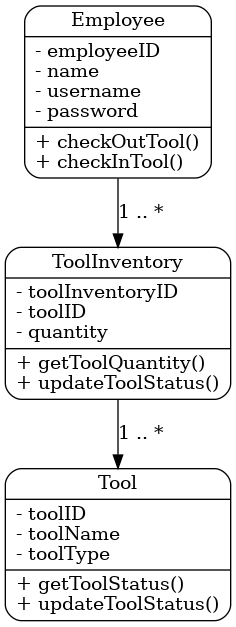
|  |  |  |
| --- | --- | --- |
| Entity/Component | Type | Description |
| Employee | Class | id, name, username, password\_hash/role; ops: login(). |
| Tool | Class | id, barcode, condition, current\_employee\_id?; ops: setCustodian(), clearCustodian(). |
| Transaction | Class | id, employee\_id, tool\_id, timestamp, action{CHECK\_OUT,CHECK\_IN}. |
| Warehouse | Class | id, location. |
| Supply | Class | id, quantity, warehouse\_id; consumables, not checked out. |
| AuthService | Service | login(), session mgmt. |
| ToolService | Service | getToolByBarcode(), updateCondition(). |
| TransactionService | Service | checkout(), checkin() with validation + atomic writes. |
| ReportingService | Service | status/history reports. |
| Repositories | DAO | CRUD for each entity with parameterized queries. |

# 5. Component Design

## 5.1 Static View (Class Responsibilities)

Key relationships: Employee 1..\* Transaction; Tool 1..\* Transaction; Tool 0..1 → Employee (current custodian). Services depend on Repositories; UI depends on Services.

Figure: UML Class Diagram (current).



## 5.2 Dynamic View (Key Scenarios)

Checkout and Check-in sequences:

Figure: Sequence Diagram — Checkout.

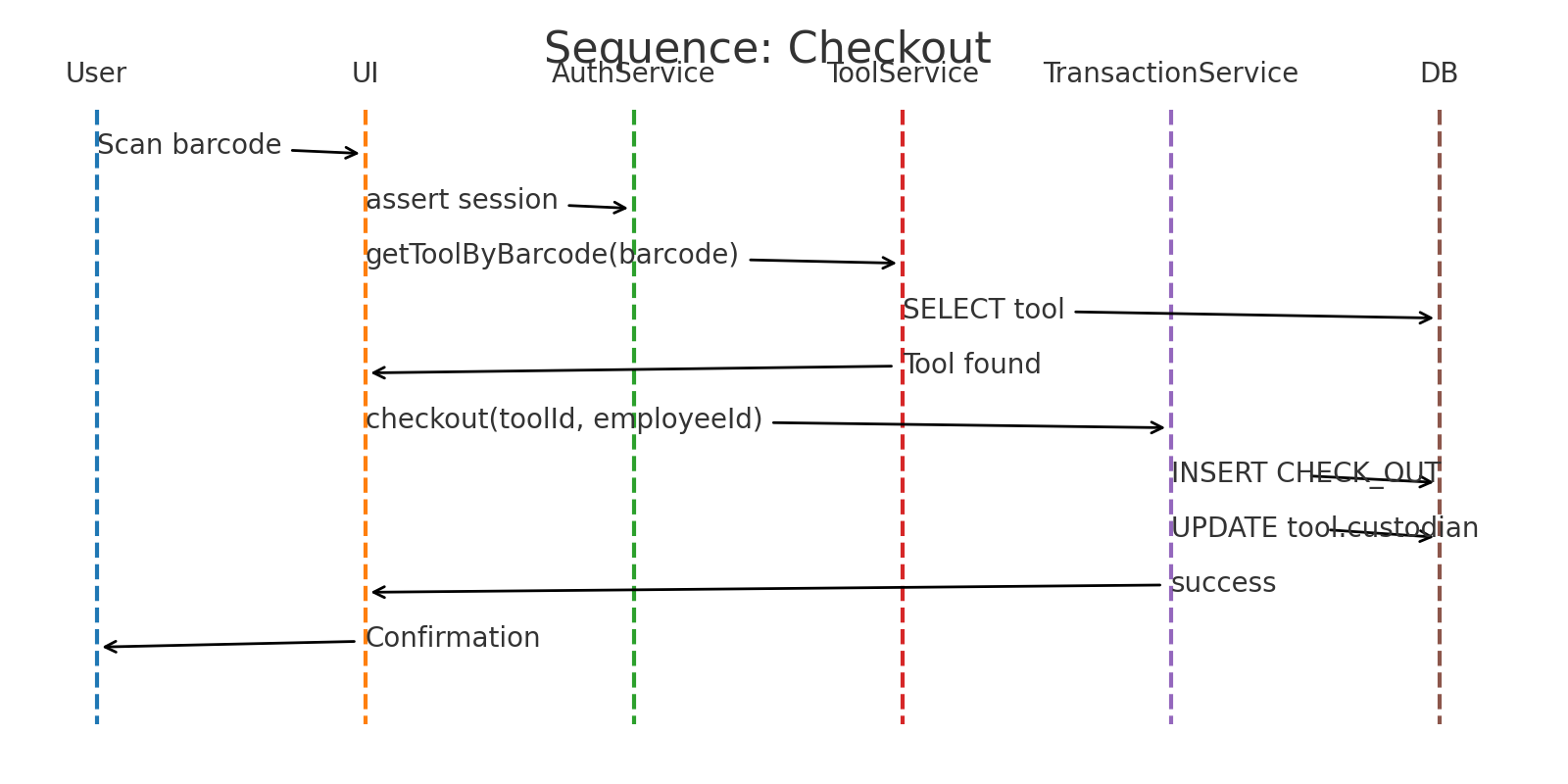
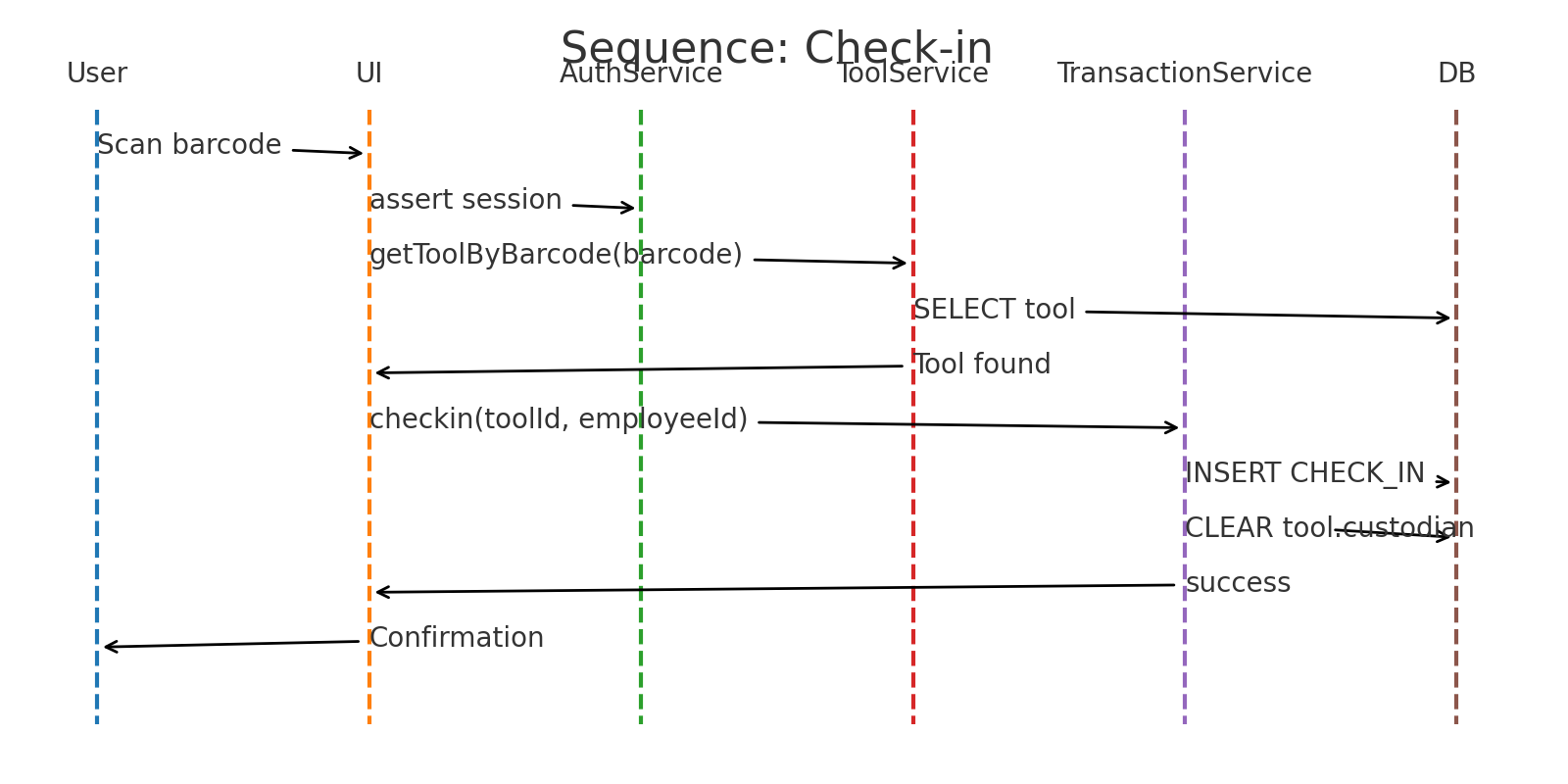


Figure: Sequence Diagram — Check-in.



## 5.3 Design Patterns

Singleton (DB/RepoFactory), Factory Method (ScannerAdapterFactory), Observer (status updates to UI).

## 5.4 Interfaces

AuthService.login(username, secret)->Session  
ToolService.getToolByBarcode(bc)->Tool  
ToolService.updateCondition(id,cond)  
TransactionService.checkout(toolId,empId)->Transaction  
TransactionService.checkin(toolId,empId)->Transaction

## 5.5 Error Handling & Logging

Domain errors: ToolNotAvailable, NotCustodian, UnknownBarcode. All writes are transactional; audit logs kept for auth and transactions.

## 5.6 Security Considerations

Hash+salt passwords; least-privilege DB user; parameterized queries; roles Employee/Admin.

# 6. Human Interface Design

Screens: Login, Home, Scan, Tool Details, My Tools, Transactions/History, Reports. Kiosk-friendly, minimal steps.

# 7. Requirements Traceability Matrix (Updated)

|  |  |  |  |
| --- | --- | --- | --- |
| Req ID | Requirement | Design Element(s) | Verification |
| FR-01 | Authenticate employees via credentials at kiosks/tablets. | AuthService; Login UI; EmployeeRepository | TC-AUTH-01 |
| FR-02 | Check out tools by scanning barcodes and confirming action. | ScannerAdapterFactory; ToolService; TransactionService.checkout() | TC-CHKOUT-01 |
| FR-03 | Check in tools by scanning barcodes and confirming action. | ScannerAdapterFactory; ToolService; TransactionService.checkin() | TC-CHKIN-01 |
| FR-04 | Record tool condition at check-out and check-in. | ToolService.updateCondition(); Transaction metadata | TC-COND-01 |
| FR-05 | Maintain transaction history (employee, tool, timestamp, action). | TransactionRepository; Logging | TC-AUDIT-01 |
| FR-06 | Supervisors/Managers view tools in possession by any employee. | ReportingService; Admin UI | TC-REPORT-01 |
| FR-07 | Employee views tools currently assigned to them. | ToolService; My Tools UI | TC-MYTOOLS-01 |
| NFR-01 | Performance: Common actions <= 1s. | Optimized queries; async I/O | PT-RESP-01 |
| NFR-02 | Availability during all shifts. | Deployment/DB config; monitoring | AVAIL-OBS-01 |
| NFR-03 | Usability: kiosk-first flow. | UI guidelines | UX-HEUR-01 |
| NFR-04 | Reliability: atomic transactions; no crashes. | DB transactions; error handling | REL-RESIL-01 |

# Appendix A — Coding Conventions

Naming standards; input validation; parameterized queries; no secrets in code.

# Appendix B — Open Issues

Confirm SRS IDs; finalize roles; decide on exact hardware scanner models.

# Appendix C — Updated Sequence Diagrams (replaces Section 5.2 figures)

These diagrams supersede the earlier 5.2 images for readability and consistency.

Figure C-1: ECS Sequence — Checkout (Updated)

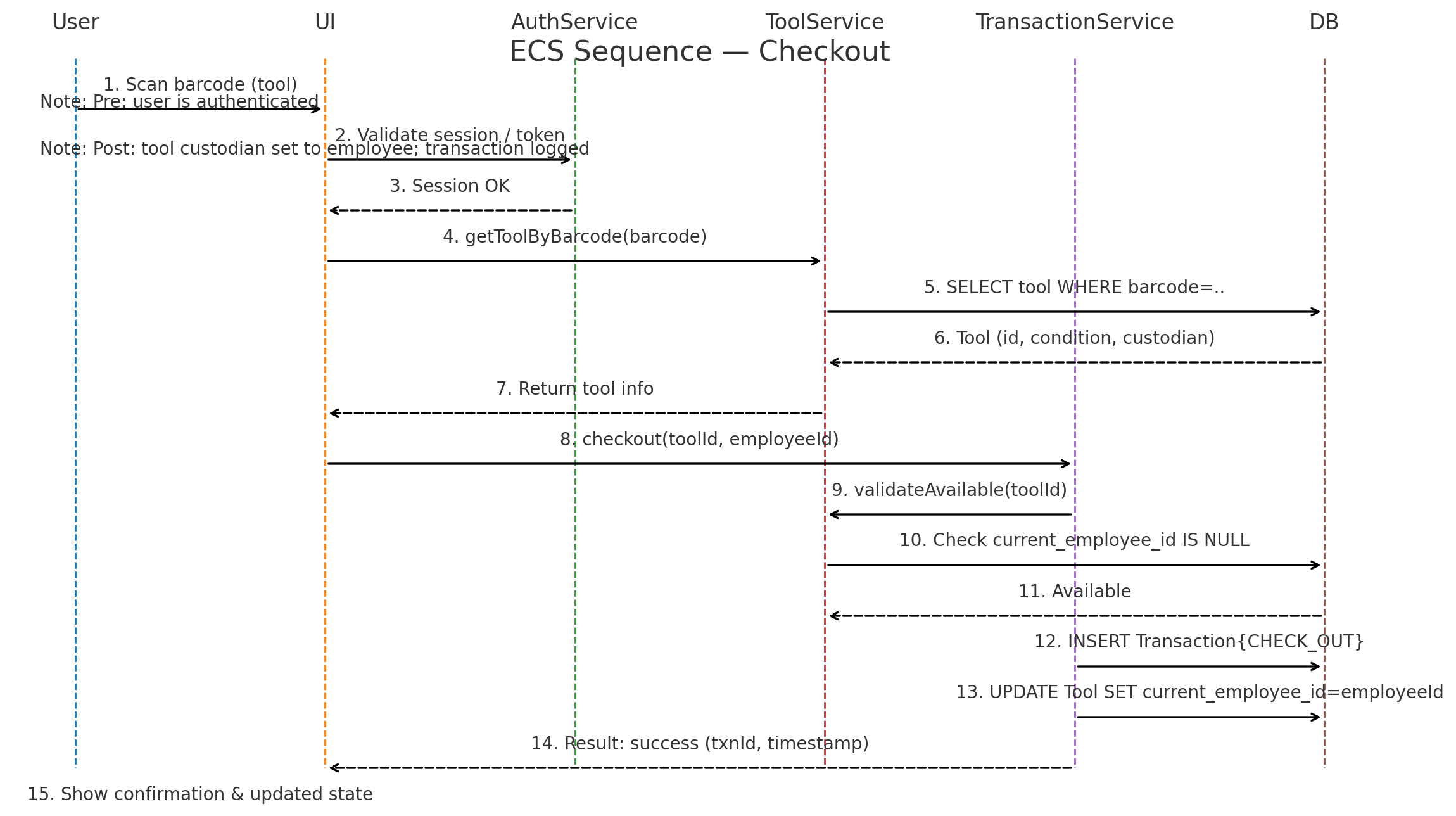


Figure C-2: ECS Sequence — Check-in (Updated)

