Tickora Attendance Management

Clock In / Clock Out Flow – Multiple Sessions

---

1. Purpose

This document describes the Clock In / Clock Out workflow in Tickora with support for **multiple work sessions per day**, focusing on system flow, rules, and data handling.

---

2. API & Service Flow

Request Flow

```
Client Application
   ↓
FaceController (/clockInOut API)
   ↓
TimesheetFacade
   ↓
FaceServiceImpl.clockInOutUser()
   ↓
TimesheetService.processTimesheetLogs()
   ↓
Database (timesheet & timesheet_history)
```

---

3. Pre-Save Validations

Before saving any attendance record, the following validations are performed: - User is active and eligible - Location validation (approved location) - Face verification via external Python service

Only after successful validation does attendance processing continue.

---

4. Attendance Model Overview

timesheet_history (Source of Truth)

- Stores **all attendance events**:

• CLOCK_IN
• CLOCK_OUT
• Each record contains:
• Timestamp
• Location
• Action source (user/system/manual)
• Supports **unlimited sessions per day**

timesheet (Daily Summary)

• One record per user per day
• Stores **calculated summary data only**:
• First clock-in (earliest)
• Last clock-out (latest)
• Total working hours
• Total break hours
• Overtime and attendance status
• Used for reports, dashboards, and payroll

---

5. Business Rules (Multiple Sessions)

• **CLOCK_IN** is allowed only if:
• No active session exists, or

• The last action was CLOCK_OUT

• **CLOCK_OUT** is allowed only if:

• There is an open CLOCK_IN session

Invalid scenarios prevented: - Multiple CLOCK_IN actions without CLOCK_OUT - CLOCK_OUT without a preceding CLOCK_IN - Overlapping work sessions

---

6. Session Processing Logic

Clock-In Handling

• Creates a new CLOCK_IN entry in `timesheet_history`
• Opens a new work session

Clock-Out Handling

• Closes the latest open session
• Creates a CLOCK_OUT entry in `timesheet_history`

Calculations

- **Working Hours**: Sum of all CLOCK_IN → CLOCK_OUT durations
- **Break Hours**: Gaps between consecutive sessions
- Summary values are continuously updated in `timesheet`

---

7. Auto Clock-Out

- If a user forgets to clock out:
- The system auto-closes **only the last open session**
- Previously completed sessions remain unchanged

---

8. Backward Compatibility & Impact

- No database schema changes
- No data migration required
- Existing attendance data remains valid
- Reports, dashboards, and payroll continue to work without change
- Manual attendance edits continue to create history records

---

9. Outcome

- Supports real-world attendance behavior (breaks, field work, multiple visits)
- Scalable and maintainable design
- Clean separation between event tracking and daily summaries
- Strong foundation for future attendance enhancements