Team Project Overview

(Fall 2020)

In this section:

- Introduction
- Current Issues
- Proposed Solution

Introduction

This project involves the creation of a Time and Attendance System, intended for internal use by a local manufacturing company.

The initial scope of the project is a set of core libraries which implement the business logic needed for the system. Later, these libraries will be integrated into a Web-based time and attendance application, as well as a clock terminal control system.

(This semester, we will be working only on the core libraries.)

Background

The company has about 150 hourly employees, assigned to various departments spread out throughout its manufacturing facility. Different employees work different shifts, which start and end at different hours of the day. All are expected to clock in on time at the start of their shift, and to clock out on time at the end of their shift.

According to company policy, if an employee *clocks in too late* and/or *clocks out too early*, *fifteen minutes per incident* are deducted from their total hours for the day.

Similarly, if an employee *clocks in too early* and/or *clocks out too late*, those extra minutes at the beginning or at the end of the day should *not* be included in their total hours.

Current Issues

- The company currently has a single time clock in the front office, where all employees line up to clock in and out. Given the number of employees, this creates major traffic problems.
- The policies for late arrivals and early departures, and for early arrivals and late departures, cannot be automated by the current system. So, a secretary in the office must manually check *every* employee's time, *every* day, and deduct the appropriate number of minutes as needed.
- The current system provides no easy way to view an employee's attendance history, including the employee's rate of absenteeism over time.

Proposed Solution

The solution decided on by the company is to implement its own time and attendance system. This system must automatically compute the employees' total hours, in a way that is fair to the employees while remaining in accordance with company policy. Desired features include:

- The ability for employees to clock in and clock out from multiple clock terminals, which will be stationed across the various departments and connected to a central server.
- The ability to automatically adjust an employee's punches either forward or backward in time, for the purpose of computing their adjusted accrued hours, according to the policies described earlier.
- The ability to automatically compute the total hours accrued by the employee, within a single day *and* over an entire pay period.

Proposed Solution (cont'd)

The rules by which the employees' punches should be adjusted are described below.

Here, we use the terms "**Shift Start**" and "**Shift Stop**" to refer to the regularly scheduled starting and stopping times of the employee's shift, and "**Lunch Start**" and "**Lunch Stop**" to refer to the start and stop of the shift's scheduled lunch break.

The important increments of time (measured in *minutes*) to use for the adjustments are as follows:

- **Interval**: The number of minutes *before* the *start* of a shift, and *after* the *end* of a shift, in which an employee's early "clock in" and late "clock out" punches are adjusted *forward* to the scheduled start of their shift, or *backward* to the end of their shift, respectively.

Proposed Solution (cont'd)

- **Grace Period**: The number of minutes *after* the *start* of a shift, and *before* the *end* of a shift, in which an employee's late "clock in" punches or early "clock out" punches are "forgiven." These punches are adjusted backward to the scheduled start of their shift or forward to the end of their shift, respectively; thus, the employee is considered to have arrived or departed on time if they clock in or out "close enough" to the shift starting or stopping time.
- **Dock**: If a late "clock in" punch is made *too* late after the start of the shift to fall within the grace period, the punch is adjusted forward in time *from the start of the shift* by this amount (to discourage excess tardiness). Similarly, if an early "clock out" punch is made *too* early before the end of the shift to fall within the grace period, the punch is adjusted backward in time *from the end of the shift* by this amount.

Proposed Solution (cont'd)

The adjustment rules to be performed by the system are as follows:

- "**Shift Start"** and **"Shift Stop"**: The employee's early "Clock In" punch or late "Clock Out" punch (within the Interval before the start of the shift or after the end of the shift) is to be realigned with the starting or stopping time of the shift, respectively.
- "**Lunch Start**": The employee's "Clock Out" punch (within the lunch break) is to be realigned with the start of lunch break.
- "**Lunch Stop**": The employee's "Clock In" punch (within the lunch break) is to be realigned with the stop of lunch break.

Proposed Solution (cont'd)

- "**Grace Period**": The employee's late "Clock In" punch or early "Clock Out" punch (within the Grace Period after the start of the shift or before the end of the shift) is to be realigned with the starting or stopping time of the shift, respectively.
- "**Dock**": The employee's late "Clock In" punch (within the Dock interval after the start of the shift *and* outside the Grade Period) is to be adjusted forward in time *from the start of the shift* by this amount. Similarly, the employee's early "Clock Out" punch (within the Dock interval before the end of the shift *and* outside the Grade Period) is to be moved backward in time *from the end of the shift* by this amount.

Proposed Solution (cont'd)

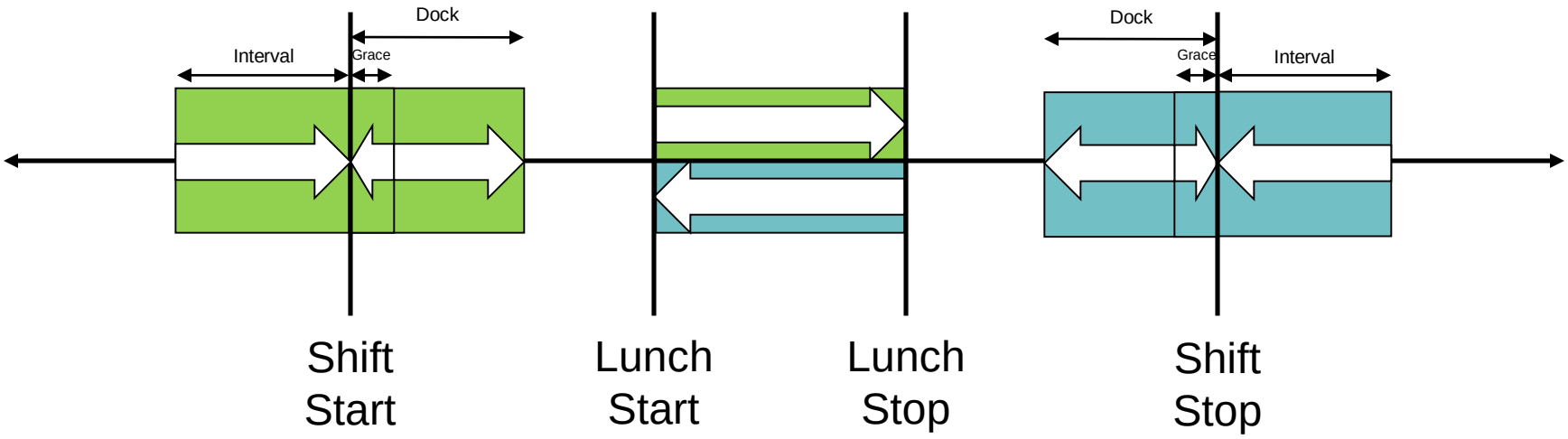
If an employee's punch falls outside any of the rules outlined above—that is, if it is outside the Intervals before the start of the shift and after the end of the shift, if it does not fall within the Grace or Dock periods at the start or end of the shift, and if it does not occur within the designated lunch break—it should be *rounded up or down* to the *nearest increment* of the Interval.

If the punch happens to occur at an even increment of the Interval, disregarding the seconds—for example, if the Interval is set to 15 minutes and a first-shift employee clocks in at exactly 9:30—then no adjustment is necessary.

Proposed Solution (cont'd)

Adjustment of "Clock In" Punches

Adjustment of "Clock Out" Punches



The punch adjustment rules, as depicted on a timeline. (The respective durations of the shift, lunch break, and intervals are not shown to scale.)

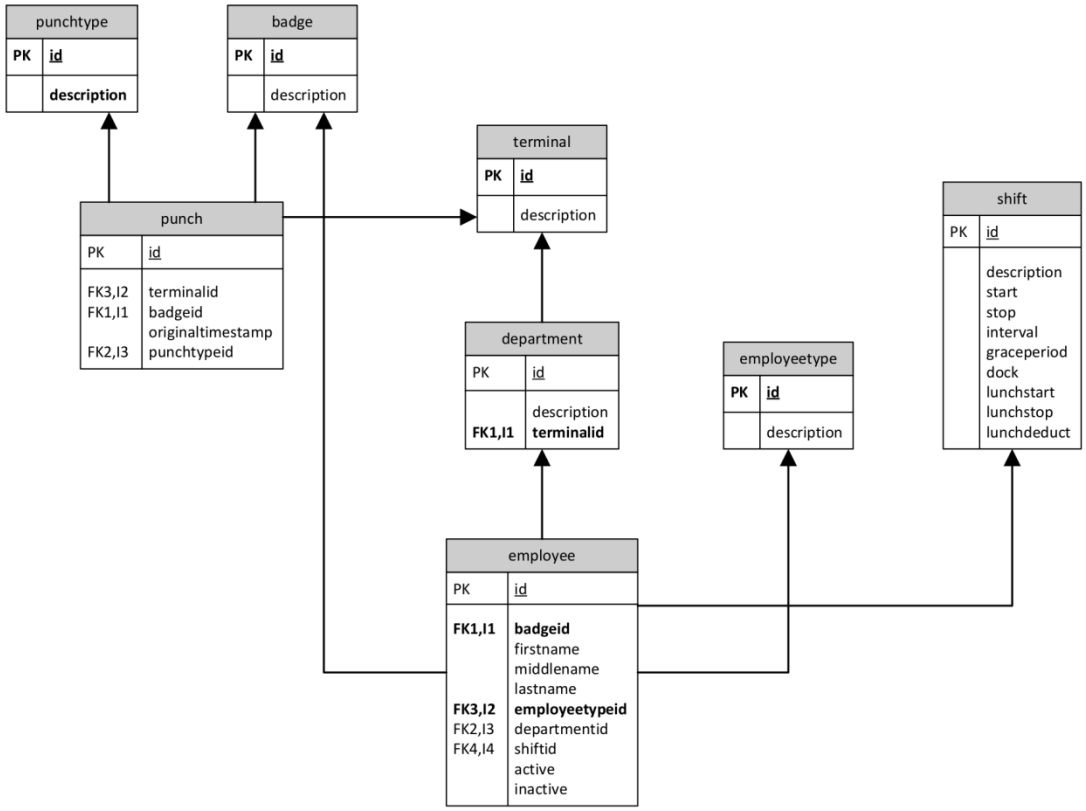
Proposed Solution (cont'd)

Before building the entire system, it is first necessary to design and implement the database and the corresponding business logic. This will be the main focus of the team project.

An initial database design (depicted on the next page) has already been implemented. A large sample of punches from the old system, accumulated over a few months in 2018, has been imported for your use during development and testing.

Your main focus will be on writing the code which will read and write the database objects in a way that abstracts away the underlying database implementation. Once these objects can be created and exchanged within the application, you will be ready to implement the business logic.

Proposed Solution (cont'd)



The Time and Attendance Database