Software Design Specification

Prepared by: Madeline Cohen, Josh Sindelar, and Spencer Van Rossum

October 9, 2025

System Description

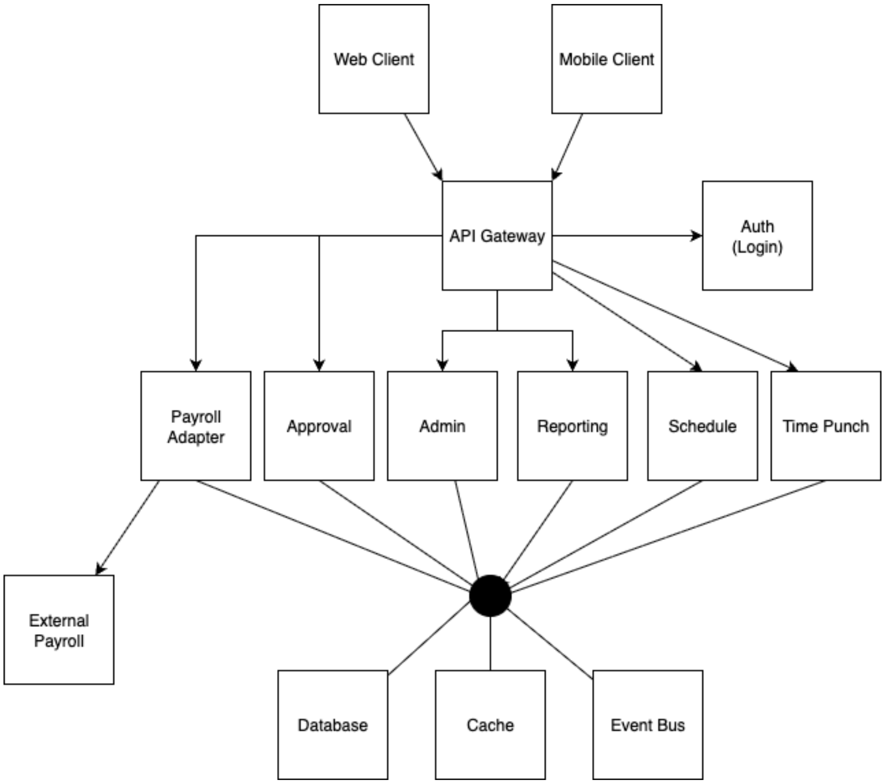
The Employee Time Tracking System is a software application designed to manage and record employee work hours accurately and efficiently. The system allows employees to “clock in” and “clock out” through a secure user interface, while the admin can monitor attendance, schedules, and reports on work hours and overtime. The main goal of the system is to replace manual timekeeping methods with an efficient, automated, and reliable solution.

The system will consist of three major user types: employees, managers, and administrators.

- Employees can clock in and out, view their current week’s hours, and request time off.
- Managers can view attendance, approve or adjust time punched, and manage employee schedules.
- Administrators can manage user accounts, system settings, and generate company wide reports.

Software Architecture Overview

- Architectural diagram of all major components



Web/Mobile Client

- Single-page app or website for clock-in/out, schedules, approval, reports, admin, and settings

API Gateway

- The hub for all incoming client requests. It ensures that clients never directly access internal services or databases, improving both security and maintainability.

Authentication (Login)

- Handles user authentication and authorization. It verifies user credentials, issues tokens, and enforces role-based security. Employees, managers, and admins have different permissions, and Auth ensures that users only access data and functions appropriate for their role.

Services

- Time Punch allows users to clock in/out. It records timestamps and calculates work duration.
- Schedule allows administrators to assign shifts and review attendance.
- Reporting combines data from time punches and schedules to generate summaries, reports, and overtime calculations.
- Admin provides administrative functions for users designated as administrators.
- Approval allows administrators to review, approve, or correct employee time records.
- Payroll Adapter acts as a bridge between the internal system and External Payroll.

External Payroll

- An external payroll system that receives reports from the Payroll Adapter. It allows automated payroll processing without manual re-entry of hours.

Database

- The database holds all persistent records such as user accounts, schedules, time punches, approvals, and reports.

Cache

- Cache provides temporary storage for frequently accessed information such as current user sessions, active punches, or recent reports.

Event Bus

- The Event Bus is an asynchronous communication mechanism that allows services to publish and subscribe to system events.

Web/Mobile Client → API Gateway

- All user actions are securely sent via API requests.

API Gateway → Auth (Login)

- Handles authentication and validates tokens for every request.

API Gateway → Services

- Routes user requests to the appropriate microservice.

Services ←→ Database

- Each service stores and retrieves its data in the shared database.

Services ←→ Cache

- Used for fast lookups of frequently requested data.

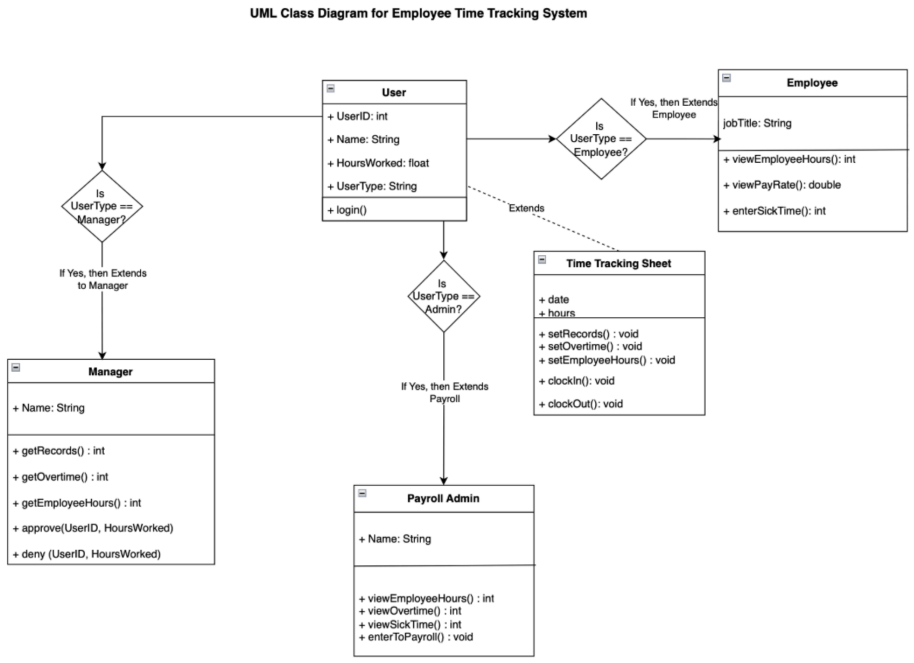
Services ←→ Event Bus

- Services broadcast events (like “time punch recorded” or “report generated”) to keep the system synchronized.

Payroll Adapter → External Payroll

- Transfers processed, approved work data to payroll systems for payment.

- UML Class Diagram:



- Description of classes - The main class is the User class, which extends other classes' operations depending on authorization which is determined by UserType. The User class is meant as a foundation for other classes which define the type of employee accessing the time tracking system. For example, the different types fall under the Payroll Admin, Manager, and Employee classes. The Manager class has operations that help manage other employees, the Payroll Admin class contains methods to handle payment and the Employee class is designed for any generic employee that utilizes the time tracking system. The last class included is the Time Tracking Sheet class which is able to be utilized by any class and includes the main functionality of the system such as clocking in and out.
- Description of attributes - The attributes listed in the User class includes an int UserID, a String Name, a float HoursWorked, and a String UserType. The Time Tracking Sheet includes the date and hours attributes which are float values. These attributes mainly describe the specific user accessing the time tracking system allowing for individualized tracking to make sure that each employee gets paid for exactly what they worked, one employee can't clock in another employee, and certain employees not having authorization to manager or payroll admin methods when they aren't supposed to.
- Description of operations - The main User class has one operation, which is login(). The parameters for this operation is the UserID and a preset password. Once a user is logged in, they have access to the Time Tracking Sheet operations which include setRecords(), setOvertime(), setEmployeeHours(), clockIn(), and clockOut(). Each of these methods are of void return type and the first three methods are designated for administration and the parameters are the UserID of the specific Employee, and a float value for specific overtime and employee hour values. The clockIn and clockOut operations are the most utilized functions in the System, with no parameters. They are meant to iterate the hours the specific Employee works and then add the amount they worked into their hours attribute. For the Employee class, they can viewPayRate() which is a generic get function that returns a double type. They can also access the enterSickTime() method which allows them to enter an int of how much sick time they would like to use. For the Manager class, there are a lot of get

functions that return what is in the operation name, such as getRecords(), getOvertime(), and getEmployeeHours(). They also have approve and deny operations with UserID and HoursWorked parameters. Lastly the Payroll Admin class has get functions which are viewEmployeeHours(), viewOvertime(), viewSickTime(), and then they have an enterToPayroll operation which allows the Administrator to enter specific employees and their hours worked to confirm the payment to said employee.

Development Plan and Timeline

Partitioning of Tasks

- Requirements Review and Design Finalization (Week 1-2, from project start)
    - Review requirements specification and finalize the UML and architecture diagrams.
- Backend Development (Week 3-5)
    - Build database and core system functions.
    - Create APIs for login, time punches, and reports.
- Frontend Development (Week 6-8)
    - Design user interfaces for employee, supervisor, and admins.
    - Connect frontend to backend.
- Testing and Integration (Weeks 9-10)
    - Combine all system parts.
    - Perform testing and fix issues found.
- Deployment and Documentation (Weeks 11-12)
    - Deploy the final system.
    - Complete documentation and prepare presentation.

Team Member Responsibilities

- Madeline Cohen- Project coordination, requirements review, documentation, task scheduling, and timeline management

- Josh Sindelar- Software architecture design, backend API development, and database integration.
- Spencer Van Rossum- UML class diagram creation, class and method documentation, and frontend implementation.