



FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

College of Computer Studies

CS SPECIALIZATION 2 - Programming Tools and Techniques
(CS0053)

Employee Management System
Case Study 1

Submitted by:

Jimeno, Eymard Julian

Lobaton, Reannah Ruth

Murayama, Vincent Karl

Ordoñez, Kendric

Submitted to:

MR. HADJI TEJUCO
Professor

02 October 2023

Table of Contents

Table of Contents.....	2
I. Introduction:.....	3
Purpose.....	3
Scope.....	3
Definitions.....	3
II. Overall Description.....	5
Product Perspective & Features.....	5
User Classes and Characteristics.....	5
Operating Environment & Design Constraints.....	6
Assumptions and Dependencies.....	7
Functional Requirements.....	9
User Record/Profile Editing.....	9
Admin Menu.....	10
Unpaid/Paid Leave Application Form.....	11
Access Company Resources.....	12
Attendance and Overtime Records.....	13
Non-Functional Requirements.....	14
User Interface Requirements.....	14
Hardware and Software Requirements.....	14
Data Management or Database Requirements.....	14
System Models.....	14
Application Entity Relationship Diagram.....	14
Testing.....	15
Screenshots.....	15

I. Introduction:

Purpose

TechSolve is an IT services firm with a global presence and has over 5,000 employees spread across multiple locations. However, challenges were introduced due to the increasing scale of the company, and its manual HR processes. This challenge highlights the issues of outdated management systems that rely on manual processes.

Scope

To address this, the group developed a lightweight CLI application using C++; the application is called Employee Management System. The application is made to provide a streamlined process for onboarding, resource management, and employee management. The main dashboard showcases the following features of the application: User/Employee Data Update form, Unpaid/Paid Leave Application Form, User/Admin Resources, and User Records.

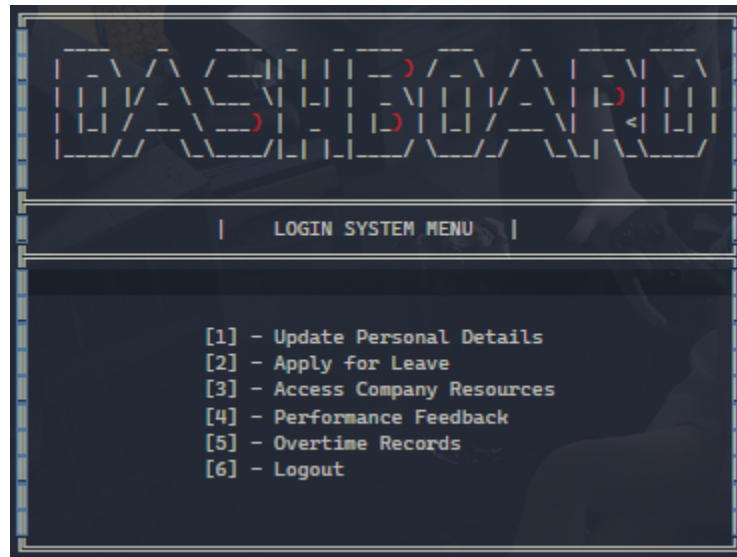
Definitions

- IT: Information Technology. It refers to the use of computers and telecommunications equipment to store, retrieve, transmit, and manipulate data.
- HR: Human Resources. This department manages various aspects related to employees, including recruitment, training, performance evaluations, and payroll.
- CLI: Command-Line Interface. A user-interface to computer software where the user interacts by typing commands directly.
- C++: A high-level programming language that's commonly used for system/application software, game development, drivers, and embedded firmware.
- GUI: Graphical User Interface. It allows users to interact with a system through graphical icons and visual indicators.
- TIN#: Tax Identification Number. It's an identification number used for tax purposes.
- PhilHealth: The Philippine Health Insurance Corporation. It's a government agency that ensures the health of the Filipino people.
- SSS: Social Security System. A social insurance program in the Philippines for workers in the private, professional, and informal sectors.
- JSON: JavaScript Object Notation. A lightweight data-interchange format that's easy for humans to read and write and easy for machines to parse and generate.
- Microsoft Visual C++: An integrated development environment (IDE) product from Microsoft for the C, C++, and C++/CLI programming languages.
- nlohmann (Niels Lohmann): Refers to a popular open-source JSON library for C++ developed by Niels Lohmann.
-

- Boolinq: A library in C++ that brings SQL-like query abilities.
- SQL: Structured Query Language. It's a domain-specific language used in programming for managing and querying data held in a relational database management system.
- Entity Relationship Diagram (ERD): A graphical representation of the logical structure of a database. It shows the relationships between entities (tables).
- Admin: Short for "administrator." It often refers to a user with elevated rights or permissions in a software system.
- Windows 10 and Windows 11: Operating systems developed by Microsoft. They're the successors to Windows 8.1 and Windows 10, respectively.
- OS: Operating System. Software that communicates with the hardware and allows other programs to run.
- Performance Appraisal: A regular review of an employee's job performance and overall contribution to the company.
- Onboarding: The process of integrating a new employee into the organization and its culture.
- Business Performance Management: The process of analyzing business performance and taking steps to improve it.

II. Overall Description

Product Perspective & Features



The integration of the Employee Management System for TechSolve modernizes and streamlines TechSolve Inc.'s HR processes to enhance administrative efficiency, improve employee satisfaction, and support the company's continued growth.

The application provides multiple company management tools that are related to their business operations which is developed to streamline the workflow by mitigating the time consumed by said manual processes. The application has utilities/tools for HR processes like Employee engagement, Performance Appraisal, Onboarding, Business Performance management and also some additional features are provided in the application. It was also made with modularity in mind which is why all features are subdivided into their own functions for readability and scalability.

User Classes and Characteristics

The application is intended to be used daily as a utility/tool for management making it an essential component of company operations, the expected primary users for this application are company administrators and employees giving them certain levels of access according to their roles. Administrator accounts have full control and access over all application features and resources giving them access to features for managing their respective department/division, and lastly default accounts/employee accounts are given access to some application features and tools for the purpose of streamlining company processes.

Operating Environment & Design Constraints

The program only supports compatibility for Windows operating systems, specifically updated Windows 10 and Windows 11 systems, this is due to limitations in available libraries that were used during its development. It should be considered that the application also has limited functionality due to the interface being the Windows CLI. It should also be noted that a large-scale company should have a managed shared database system for this application where users have access to all the files that the application needs; as mentioned earlier, the application is only limited to mitigate possible time loss caused by manual processes and is intended to run alongside the operating system; and is not intended to replace the operating system itself; to elaborate, the developers cannot extend the application's functionality for some features since the CLI cannot interact with specific files types; although possible, since the application is intended to run on company-provided devices using windows 10 as its operating system, replacing operating systems with a CLI based application would only hinder the workflow performance.

```
1 //
2 // [ ] [ ] [ ] [ ] [ ] JSON for Modern C++
3 // [ ] [ ] [ ] [ ] [ ] version 3.11.2
4 // [ ] [ ] [ ] [ ] [ ] https://github.com/nlohmann/json
5 //
6 // SPDX-FileCopyrightText: 2013-2022 Niels Lohmann <https://nlohmann.me>
7 // SPDX-License-Identifier: MIT
8
9 /*****
10  * Note on documentation: The source files contain links to the online
11  * documentation of the public API at https://json.nlohmann.me. This URL
12  * contains the most recent documentation and should also be applicable to
13  * previous versions; documentation for deprecated functions is not
14  * removed, but marked deprecated. See "Generate documentation" section in
15  * file docs/README.md.
16  *****/
17
18 #ifndef INCLUDE_NLOHMANN_JSON_HPP_
19 #define INCLUDE_NLOHMANN_JSON_HPP_
20
21 #include <algorithm> // all_of, find, for_each
22 #include <cstdint> // nullptr_t, ptrdiff_t, size_t
23 #include <functional> // hash, less
24 #include <initializer_list> // initializer_list
25 #ifndef JSON_NO_IO
26 #include <iosfwd> // istream, ostream
27 #endif // JSON_NO_IO
28 #include <iterator> // random_access_iterator_tag
29 #include <memory> // unique_ptr
30 #include <numeric> // accumulate
31 #include <string> // string, stoi, to_string
32 #include <utility> // declval, forward, move, pair, swap
33 #include <vector> // vector
34
35 // #include <nlohmann/adl_serializer.hpp>
36
37 // [ ] [ ] [ ] [ ] [ ] JSON for Modern C++
38 // [ ] [ ] [ ] [ ] [ ] version 3.11.2
39 // [ ] [ ] [ ] [ ] [ ] https://github.com/nlohmann/json
40 //
41 // SPDX-FileCopyrightText: 2013-2022 Niels Lohmann <https://nlohmann.me>
42 // SPDX-License-Identifier: MIT
43
44
45 #include <utility>
46
47 // #include <nlohmann/detail/abi_macros.hpp>
48
49 //
50 // [ ] [ ] [ ] [ ] [ ] JSON for Modern C++
51 // [ ] [ ] [ ] [ ] [ ] version 3.11.2
52 // [ ] [ ] [ ] [ ] [ ] https://github.com/nlohmann/json
53 //
54 // SPDX-FileCopyrightText: 2013-2022 Niels Lohmann <https://nlohmann.me>
55 // SPDX-License-Identifier: MIT
56
57
58 // This file contains all macro definitions affecting or depending on the ABI
59
60 #ifndef JSON_SKIP_LIBRARY_VERSION_CHECK
61 #if defined(NLOHMANN_JSON_VERSION_MAJOR) && defined(NLOHMANN_JSON_VERSION_MINOR) && defined(NLOHMANN_JSON_VERSION_PATCH)
62 #if NLOHMANN_JSON_VERSION_MAJOR != 3 || NLOHMANN_JSON_VERSION_MINOR != 11 || NLOHMANN_JSON_VERSION_PATCH != 2
63 #warning "Already included a different version of the library!"
64 #endif
65 #endif
66 #endif
67 #endif
```

Assumptions and Dependencies

- **Limited Compatibility:** The application is only compatible with specific Windows versions (Windows 10 and Windows 11), which could pose challenges for employees using different operating systems.
- **Interface Complexity:** The use of a Command Line Interface (CLI) may not be user-friendly for all employees, potentially leading to a learning curve and decreased efficiency.
- **Functionality Constraints:** The CLI-based application has limited functionality and cannot interact with specific file types, which may result in incomplete or inefficient HR processes.
- **Dependency on Shared Database:** The reliance on a managed shared database system introduces a single point of failure. If the database encounters issues or goes down, it could disrupt operations.
- **Data Security:** Ensuring the security of sensitive HR and company data within the shared database is crucial. Any breach or data loss could have significant consequences.
- **Limited Cross-Platform Support:** The application's Windows-only support may pose challenges for employees using other operating systems, potentially causing inconsistencies in HR processes.
- **Maintenance Challenges:** Supporting and maintaining a CLI-based application may require specialized skills and could be more complex than maintaining a user-friendly graphical interface.
- **Resistance to Change:** Employees accustomed to manual processes may resist transitioning to the new system, affecting adoption rates and potentially causing delays.
- **Workflow Disruption:** Attempting to replace the Windows operating system with a CLI-based application on company-provided devices could disrupt existing workflows and negatively impact productivity.
- **Scalability Issues:** While the application was designed with modularity in mind, it may encounter scalability issues if the company experiences rapid growth or changes in HR processes.

- Compliance and Regulatory Concerns: Ensuring that the application complies with relevant regulations and industry standards, especially regarding data privacy and security, is crucial.
- User Training Requirements: Training employees to effectively use the CLI-based application may be time-consuming and may require additional resources.
- Dependency on Third-Party Libraries: The application's reliance on third-party libraries may introduce vulnerabilities or compatibility issues that need to be closely monitored and managed.
- Lack of Cross-Platform Flexibility: The Windows-centric approach may limit the company's ability to adapt to different platforms in the future, potentially hindering flexibility and growth.
- Cost Implications: Depending on the complexity of the system, there may be significant costs associated with development, maintenance, and ongoing support.

For dependencies, the application uses some default preprocessors and libraries provided by the Microsoft Visual C++ runtime package, the application utilizes an open source JSON library by nlohmann (Niels Lohmann) as the primary input/output handler. Additionally, the application also uses Boolinq; a library which implements the use of SQL-like statements into C++ functions for data structures.

```
namespace boolinq {
    namespace priv {
        // The result_of was removed since C++20 by not all but some compilers.
        // For the sake of compatibility, use own define but in private subspace
        // to avoid collisions with std in case of using both std and boolinq.
        template<typename _Callable>
        struct result_of;

        template<typename _Callable, typename... _Args>
        struct result_of<_Callable(_Args...)> {
            typedef decltype(std::declval<_Callable>()(std::declval<_Args>()...)) type;
        };
    }

    //

    struct LinqEndException {};

    enum BytesDirection {
        BytesFirstToLast,
        BytesLastToFirst,
    };

    enum BitsDirection {
        BitsHighToLow,
        BitsLowToHigh,
    };
}
```


System Features and Requirements

Functional Requirements

User Record/Profile Editing:

Inputs: User profile data.

Processing: Editing and updating user profiles, including Name, Gender, Address, TIN, PhilHealth, SSS, and more.

Outputs: Serialized data stored in a JSON file.

Admin Menu:

Inputs: Administrator commands and user data.

Processing: Administrative control over user data, including adding, editing, and deleting employee records.

Outputs: Serialized data updated in real-time.

Unpaid/Paid Leave Application Form:

Inputs: Leave application details (start and end dates).

Processing: Recording leave applications and formatting data.

Outputs: Recorded data accessible by authorized users.

Access Company Resources:

Inputs: Performance evaluation data, assessment submissions, and feedback requests.

Processing: Performance evaluation based on various factors, including Job knowledge, Work Quality, Punctuality, Communication Skills, and Overall Rating.

Outputs: Evaluation results and feedback.

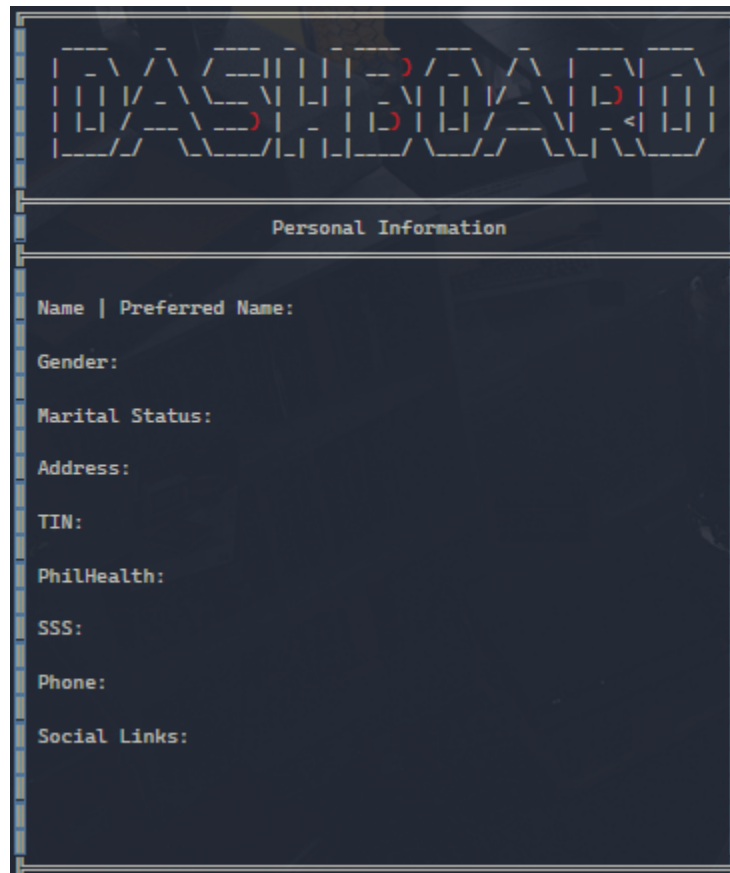
Attendance and Overtime Records:

Inputs: Employee login data, attendance records.

Processing: Tracking attendance and calculating overtime pay.

Outputs: Attendance records and overtime calculations.

User Record/Profile Editing



Personal Information

Name | Preferred Name:

Gender:

Marital Status:

Address:

TIN:

PhilHealth:

SSS:

Phone:

Social Links:

In this menu, employees can edit their profiles in the system, administrators have a higher level of access and can edit other user information for management purposes, admin accounts also have additional features for onboarding processes like adding new employee records, editing employee records, and deleting employee records. Some of the following fields to be edited and filled out are as follows; Name, Gender, Status, Address, TIN#, PhilHealth, SSS, Phone no, Social Media links, and more; the collected data is then serialized onto a JSON file which is updated real-time.

Admin Menu



As mentioned earlier, administrators have a higher level of access compared to default accounts, admin accounts can use all the default features along with 2 distinct options for employee data management; in a company setting, admin accounts are given to HR departments/managers to manage low-level company operations.

The admin accounts are also responsible for performance feedback; once requested through the system, if possible, the extended functionality of administrators receiving prompts from the application can also be implemented if specifications for network functionality was included.

Unpaid/Paid Leave Application Form

A terminal window showing a menu for leave applications. The title is "Unpaid/Paid Leave Application Form". Below the title, there are three options listed: [1] - Sick Leave, [2] - Vacation Leave, and [3] - Leave List. The interface is styled with a dark background and a light border.

This menu gives employees and others the ability to request for a sick or vacation leave by indicating the starting and end-dates, the data would then be recorded and can be accessed by authorized members, the data is then later read from the recorded JSON file and then displayed with the necessary information when being evaluated by admins.

A terminal window showing a form for entering dates. The title is "Unpaid/Paid Leave Application Form". Below the title, there is a prompt "(mm-dd-yy)". The form asks for "Enter Starting Date:" and "Enter End Date:". Below the prompts, there is a instruction "Press UP/DOWN to navigate." and two options: [1] Exit without Saving and [2] Save and Exit.

Starting Date	End Date	Type	Status
09-18-23	09-24-23	Sick	Pending

[1] Exit to Menu

The inputs are also formatted accordingly to avoid errors in data; replacing the typical method of sending an email weeks beforehand allows for flexibility in response times and also gives company employees/departments time to adjust sooner to avoid unprecedented consequences.

Access Company Resources

Employee Information

Name:

Department:

Manager:

Job Knowledge

Work Quality

Punctuality

Productivity

Communication Skills

Overall Rating:

Comments:

Use UP/DOWN to navigate.

Choose HR/Manager to Request:

[1] - exampleHr1

[2] - exampleHr2

[3] - Back

>

DATE	NAME	TIME IN	TIME OUT
09-19-23	Sir Cheng	1:00 PM	2:00 PM

In this submenu, users can request and access evaluations of their performance, submit assessments of a colleagues, and request for a performance feedback from HR teams, an employee's performance is measured through a few key factors upon evaluation; namely: Job knowledge, Work Quality, Punctuality, Communication Skills, and Overall Rating, HR members or other employees can also add comments for the evaluations.

Attendance and Overtime Records

The first screenshot shows the main menu of the application. It features a decorative header with a grid of characters. Below the header, the title "OVERTIME RECORDS" is centered. The menu options are listed as follows:

- [1] - Add Record
- [2] - View Records
- [3] - Back

The second screenshot shows the "Add Record" screen. It has the same header and title. The fields to be filled are:

- Time In:
- Time Out:
- Amount:
- Type (Holiday/Overtime/Nighttime):
- Reason:

Below these fields, a prompt says "Press UP/DOWN to navigate." At the bottom, there are two options: [1] Exit without Saving and [2] Save and Exit.

The third screenshot shows the "View Records" screen. It has the same header and title. Below the header, there is a table with the following data:

Time In	Time Out	Amount	Type	Reason
9:10am	1:00pm	5	Holiday	

At the bottom of the screen, there is a prompt: [1] Exit to Menu.

This menu provides Administrators/Management with an overview of the employees' attendance while also providing a calculation for their additional pay equivalent to their overtime hours. Administrators/Management can do this by selecting a record from the attendance list upon the request of managers or executives. Since this application is treated as an essential company utility, users are required to log-in to their accounts for their attendance. In addition, the flexibility of the C++ language offers further improvements to this system; as this can be implemented alongside other devices like; fingerprint sensors, ID scanners, or other biometric devices to make the process more streamlined.

This diagram shows the class structure of the program and its inheritance; the diagram shows each application feature and how it is subdivided into their own classes, the same goes for the sub-menus.

Testing

Screenshots

