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

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 This challenge highlights the issues of outdated management systems that rely on manual processes.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.

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,
    };
```

III. System Features and Requirements

Functional Requirements

User Record/Profile Editing

1. 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.

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

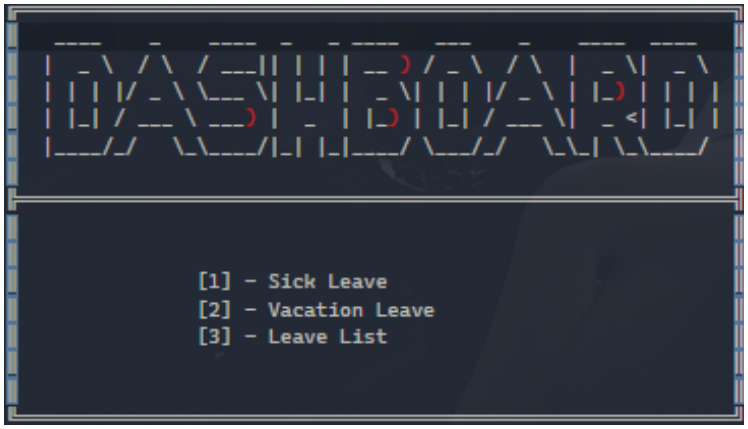
2. Admin Menu:

   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.

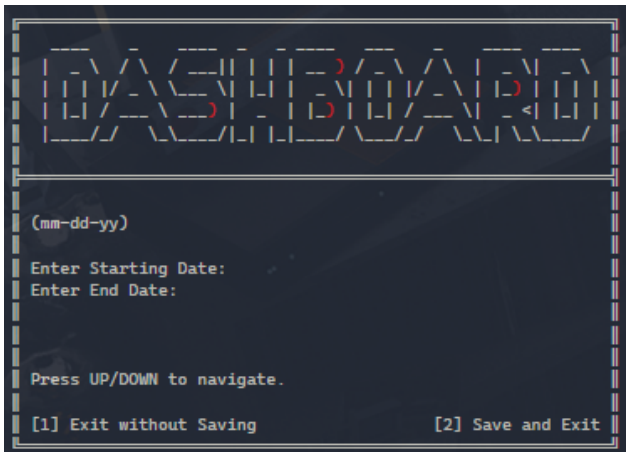
As mentioned earlier, administrators have a higher level of access compared to default accounts, admin accounts can use all the default features along with 2 distinct options for employee data management; in a company setting, admin accounts are given to HR departments/managers to manage low-level company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

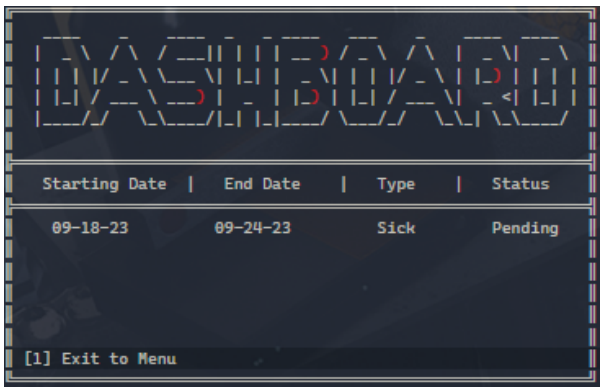
The admin accounts are also responsible for performance feedback; once requested through the system, if possible, the extended functionality of administrators receiving prompts from the application can also be implemented if specifications for network functionality was included.

Unpaid/Paid Leave Application Form



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.



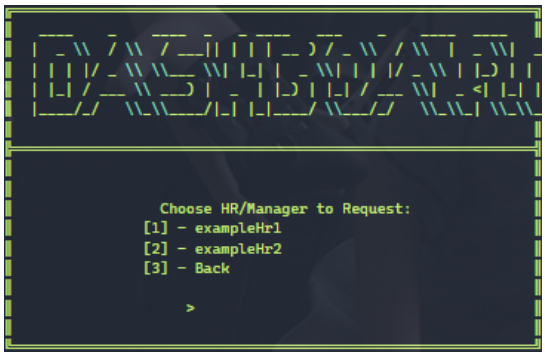


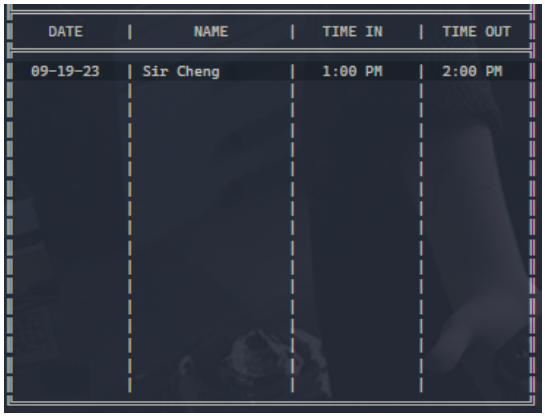
The inputs are also formatted accordingly to avoid errors in data; replacing the typical method of sending an email weeks beforehand allows for flexibility in response times and also gives company employees/departments time to adjust sooner to avoid unprecedented consequences.

3. 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



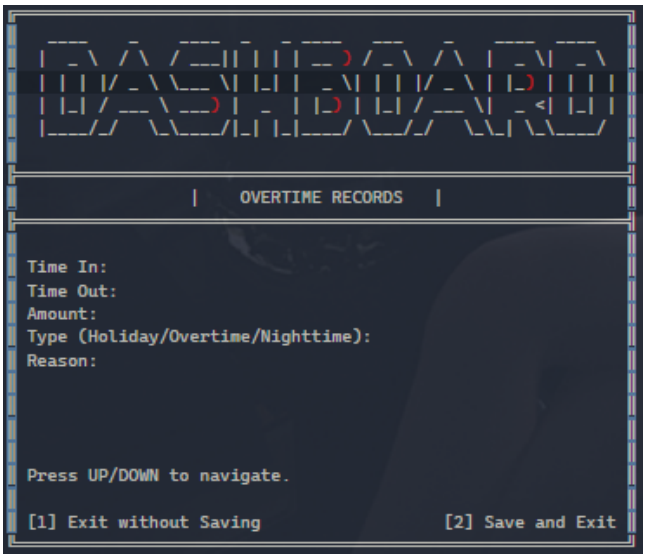


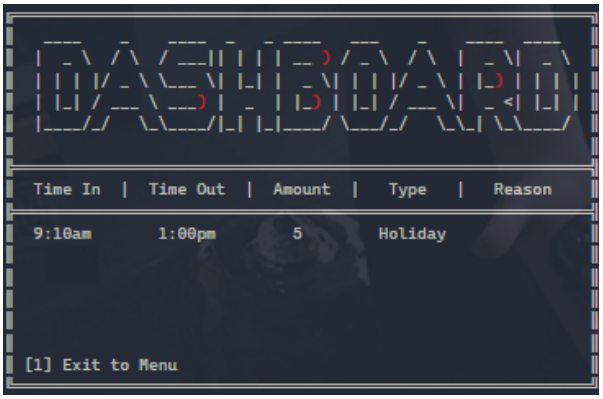
4. Access Company Resources:

   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.

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





5. Attendance and Overtime Records:

   Inputs: Employee login data, attendance records.
   Processing: Tracking attendance and calculating overtime pay.
   Outputs: Attendance records and overtime calculations.

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.

Non-Functional Requirements

- Graphical User Interface (GUI): Consider developing a graphical user interface (GUI) in addition to the CLI. This would make the system more user-friendly and accessible to employees who may find command-line interfaces challenging.

- Cross-Platform Compatibility: Expand compatibility to other operating systems (macOS and Linux) to accommodate employees using different platforms.

- Mobile Accessibility: Create a mobile-friendly version or app, allowing employees to access the system from their smartphones, which can be especially useful for leave applications and attendance tracking.

- Advanced Reporting: Implement advanced reporting and analytics features to provide insights into employee performance and HR processes. Visual dashboards can help administrators make data-driven decisions.

- Notification System: Integrate a notification system to alert employees and administrators about important updates, such as leave approval, performance reviews, and attendance discrepancies.

- Document Management: Allow users to upload and manage HR-related documents such as resumes, certificates, and performance reports securely.

- Integration with HR Software: Consider integrating the system with popular HR management software like SAP SuccessFactors, Workday, or BambooHR for seamless data synchronization.

- Enhanced Security: Strengthen data security measures, including encryption of sensitive employee data and compliance with data privacy regulations (e.g., GDPR).

User Interface Requirements

Below are descriptions of how users will interact with the CLI-based Employee Management System:

- Navigation: Users can navigate through menus by typing the corresponding numeric option and pressing enter or by using arrow keys in certain menus/sub-menus to control the position of the text-cursor.
- Inputs Types: in certain forms, users enter strings and digits which are serialized into JSON
- Data Retrieval: is done by reading and deserializing data from recorded JSON files

- Data Update/Delete: specific functionality given to both user types (administrators/employee) accounts, users can overwrite data by providing the unique account IDs to select the account to be modified or deleted.

Hardware and Software Requirements

HARDWARE REQUIREMENTS

Server Infrastructure
- A dedicated server or cloud-based server capable of hosting the application and its database. - Minimum specifications: Quad-core CPU, 8GB RAM, and 120GB SSD storage.

Database Server
- A storage system suitable for managing employee records and other related data using JSON files. - Sufficient storage capacity to handle increasing employee records and related data. - A regular backup system in place to ensure data integrity and facilitate recovery in case of failures.

Networking
- A robust network infrastructure capable of handling multiple users accessing the system simultaneously. - Firewall and other security measures to protect sensitive employee data and system resources from unauthorized access or threats.

Client Devices
Computers for administrators and other staff members to access the system. - Dual-core CPU or higher. - 4GB RAM or more. - Operating system: Windows 10 or 11. - Command-Line Interface (CLI) support.

SOFTWARE REQUIREMENTS

Operating System (OS)
- The server should operate on a stable platform such as Linux (e.g., Ubuntu Server) or Windows Server, aligned with the development team's familiarity and preference. - Client devices should run on Windows 10 or 11.

Programming Language
- C++

Development Tools
- Integrated development environment (IDE) options include Microsoft Visual C++, Code::Blocks, or other C++ supported IDEs. - Database Management System - The system will use JSON files for data storage and retrieval, leveraging the nlohmann library in C++.

Security and Encryption
- Implement strong security protocols, including user authentication and authorization mechanisms. - Secure sensitive employee data and ensure system integrity.

Version Control
- Employ a version control system such as Git for tracking changes and facilitating collaboration among the development team.

Backup and Recovery
- Schedule regular automated backups of the database and essential application files. - Develop a comprehensive disaster recovery strategy to minimize potential downtime in the event of system disruptions or failures.

Data Management or Database Requirements

DATA MANAGEMENT SPECIFICATIONS

1. Data Integrity: Ensure that data remains accurate, consistent, and unaltered during all operations. This is crucial for maintaining trust in the system.

2. Concurrency Control: Implement mechanisms to handle multiple users accessing or modifying the database simultaneously. This ensures that changes don't conflict or overwrite each other.

3. Data Redundancy Management: Minimize duplicate data to ensure efficient storage and maintain data consistency.

4. Data Scalability: The system should be designed to accommodate a growing number of records as the company expands and hires more employees.

5. Data Backup and Recovery: Regularly back up the database to secure storage locations. Implement recovery procedures to restore data in case of accidental deletions or system failures.

6. Data Security: Implement encryption techniques to secure data at rest and during transmission. This will protect sensitive employee data from potential breaches.

7. Data Archiving: As data grows, older records that are no longer actively accessed but need to be retained for compliance or other reasons should be archived.

8. Data Retention Policy: Define and implement policies for how long different types of data should be kept before they are archived or purged.

9. Data Access Control: Define user roles and permissions to determine who can access, modify, or delete specific data.

DATABASE REQUIREMENTS

1. Database Type: Given the use of JSON for data storage, a NoSQL database system would be suitable. However, if there's a need for relational data management, consider a hybrid approach.

2. Storage Capacity: The initial capacity should be based on the current number of employees and estimated growth. Consider an initial storage of at least 500GB, scalable as needed.

3. Indexing: Implement indexing on frequently accessed fields like Employee ID, Name, or Department to speed up query performance.

4. Query Performance: The database should be optimized for quick read and write operations, ensuring minimal lag for end-users.

5. Database Backup: Regularly back up the database to both local and offsite locations. Implement automatic backup solutions that capture daily, weekly, and monthly snapshots.

6. Database Recovery: Have a recovery system in place to restore the database from backups in case of failures.

7. Database Maintenance: Schedule regular maintenance tasks, including cleaning up fragmented data, updating indexes, and checking for potential issues.

8. Database Security: Implement measures such as firewalls, intrusion detection systems, and data encryption to protect the database from unauthorized access or potential threats.

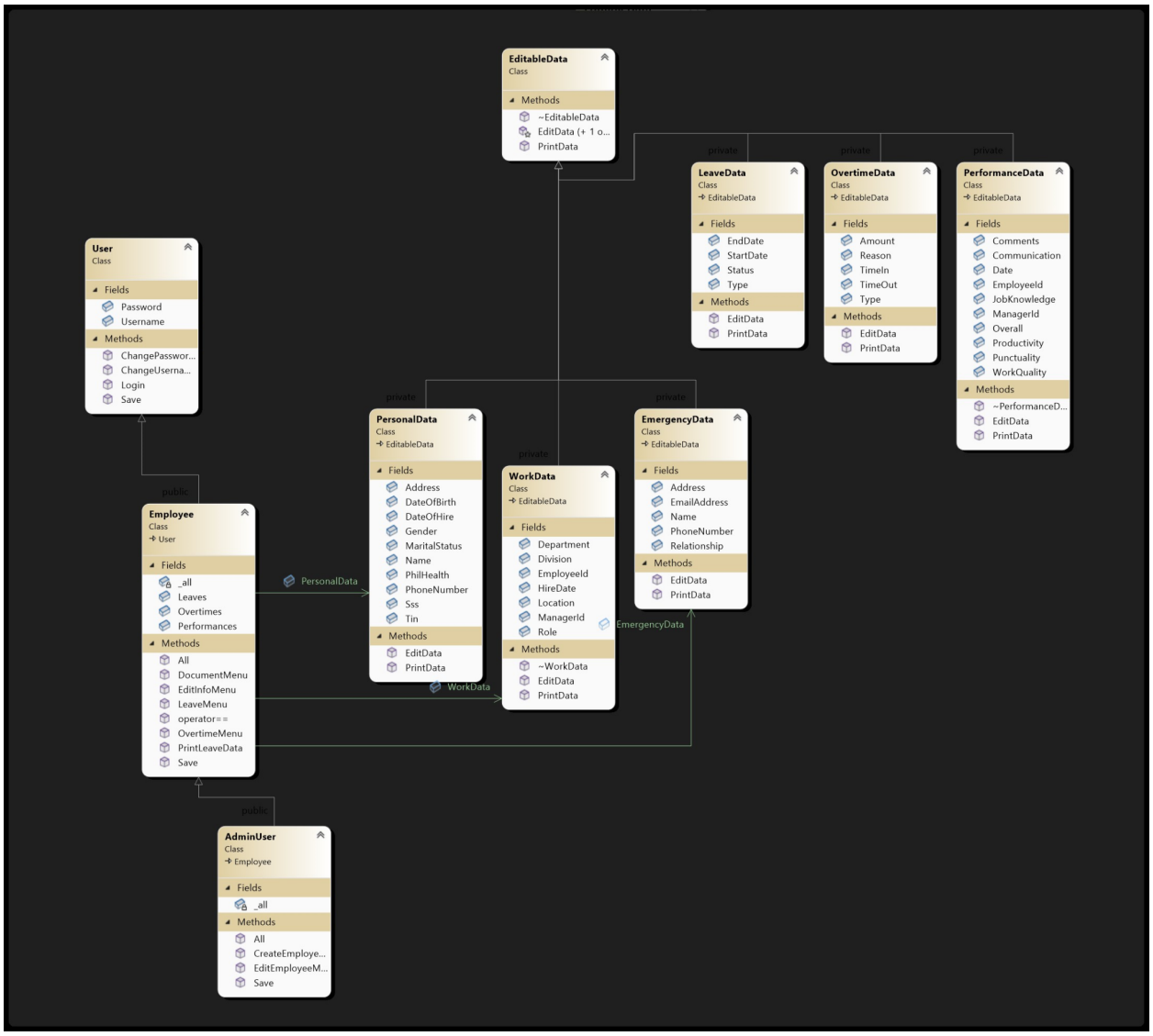
9. Database Monitoring: Utilize monitoring tools to keep an eye on database performance, usage patterns, and potential errors.

10. Database Replication: Consider database replication for distributing the database load and ensuring high availability. This is especially useful if the application is used in multiple locations.

11. Database Versioning: Track changes to the database schema, stored procedures, and other components, ensuring that upgrades or changes don't disrupt existing functionalities.

VI. System Models

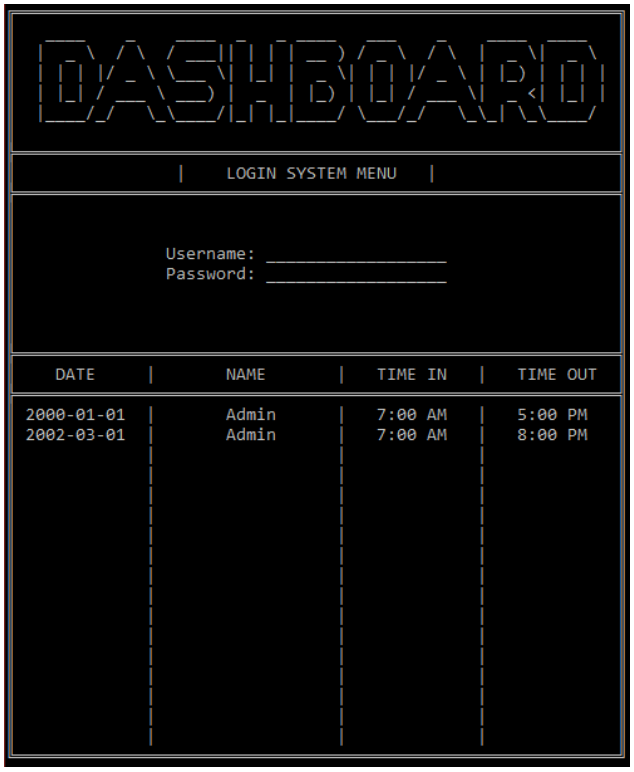
Application Entity Relationship Diagram



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.

VI. SCREENSHOTS

Login System Menu



This menu is the landing page for the program that asks users for their credentials for access. A login system is implemented for user authentication and access control which is crucial for ensuring the system's security and accessibility. This page also shows recent logins from other accounts.

Admin Menu

This dashboard shows 3 extra features that can only be accessed and used by admin accounts, the rest of the features also have a higher level of functionality; which includes access and control to all employee data, and also some other management tools; however, this system would require regulations and improved securities to avoid data manipulation schemes.

1. Create Employee - This option allows the admin to create a new employee record.
2. Edit Employee - This option lets the admin edit and make changes to the employees' personal information.
3. Delete Employee - This choice allows the admin to delete an existing employee record.
4. View Employee - This option displays the list of employees
5. Change Password - This choice lets the user to update and change their password
6. Update Personal Details - grants the employee/admin the ability to edit and update their own user profile in the system.
7. Apply for Leave - provides the employee to choose to either apply for a sick leave or vacation leave by specifying the starting and ending date.
8. Access Company Resources - displays the performance feedback by indicating the employee ID
9. Attendance Records - allows the admin/employee to add and view attendance records
10. Overtime Records - lets the admin/employee to add and view overtime records
11. Back - This choice leads the user to go back to the login system menu

Create Employee

Figure 1.1

Figure 1.2

Figure 1.3

Figure 1.1 is displayed when the user chooses the create employee option, it asks whether an Admin or Employee user type will be created. After choosing the user type, it prompts the user to input username and password (Fig. 1.3).

Figure 2.1

Figure 2.2

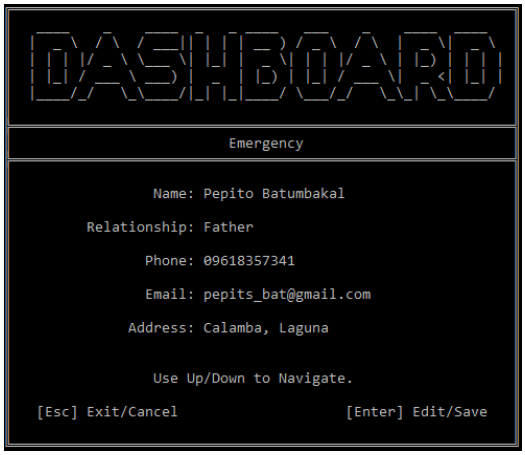


Figure 2.3

Figure 2.4

This feature allows the administrator to create employee records. Figure 2.1-3 displays the process of creating an employee. The fields to be filled out are as follows; Full legal name, Gender, Marital Status, Address, TIN#, PhilHealth, SSS, Phone no., Emergency contact, Work details, and etc. Once the employee record is created successfully, Figure 2.4 will be displayed.

Edit Employee

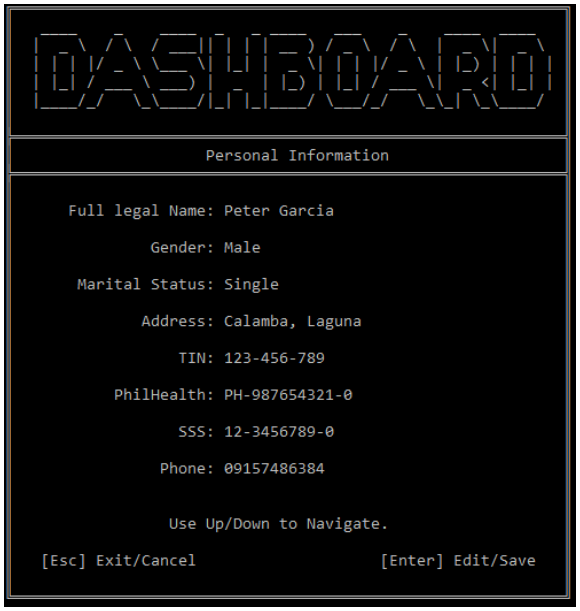


Figure 3.1

Figure 3.2

This menu allows the administrator to edit or update the employee's personal information. Figure 3.1 shows the record that was set originally, while Figure 3.2 displays the updated one.

Delete Employee

Figure 4.1

Figure 4.2

This feature allows the administrator to delete an employee record by indicating the Employee ID of the record to be deleted. Figure 4 shows that the record has been successfully deleted.

View Employee

Figure 5.1

This feature lets the user view the existing employee records. The system will ask the user to indicate the Employee ID before accessing its record.

Employee Menu

1. Update Personal Details - Ensures and helps employees keep their information up to date by being able to update.
2. Apply for Leave - Allows employees to apply for a sick or vacation leave.
3. Access Company Resources - Gives the user the option to view their performance feedback from their managers.
4. Attendance Records - Allows employees to record and view their attendance records.
5. Overtime Records - Enables the employees to log their overtime hours and view the overtime records.
6. View Employees - Allows employees easy access to other employee's information through employee ID.
7. Change Password - Allows the user to change their password.

Update Personal Details

Figure 1.1

Figure 1.2

The feature displays and allows the editing of personal information. The information displayed includes the individual's full legal name, gender, marital status, address, TIN (Tax Identification Number), PhilHealth, SSS (Social Security System), and phone.

To interact with the displayed personal information, users can navigate through the fields using the up and down arrow keys. Pressing enter selects a specific field for editing.

Figure 1.1 displays the employee's personal details that were initially set during employee creation, while Figure 1.2 reflects any changes made to the initial data.

Leave Menu

Figure 2.0

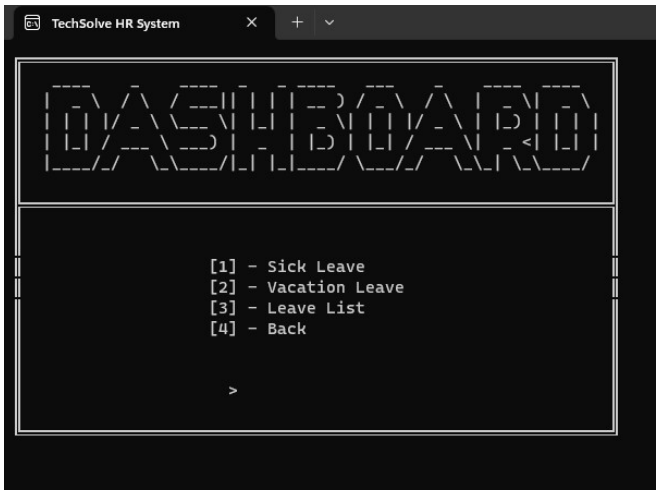


Figure 2.0 displays a menu with the following options for the 'Apply for Leave' feature:

1. Sick Leave - prompts the user to input the starting date and end date of their sick leave.
2. Vacation Leave - asks the user to provide the starting date and end date for their vacation leave.
3. Leave List - displays a list of all the set leave dates, along with the corresponding type of leave (Sick Leave or Vacation Leave).

Figure 2.1

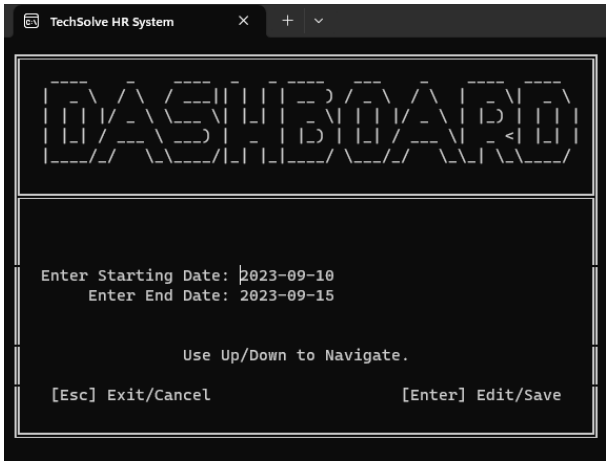


Figure 2.2

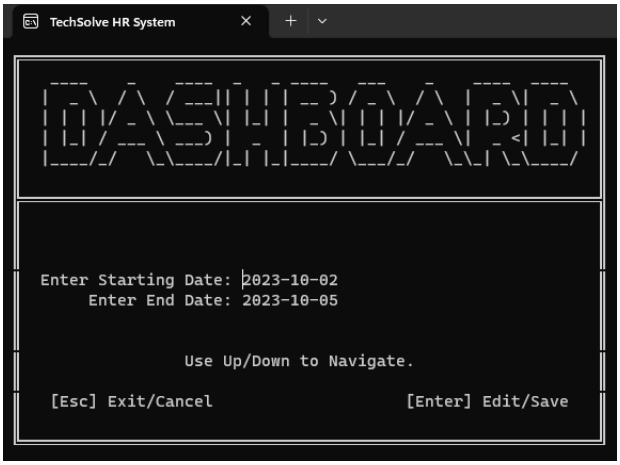


Figure 2.1 prompts the user to input the starting date and end date for Sick Leave. Figure 2.2 performs a similar function for Vacation Leave.

Figure 2.3

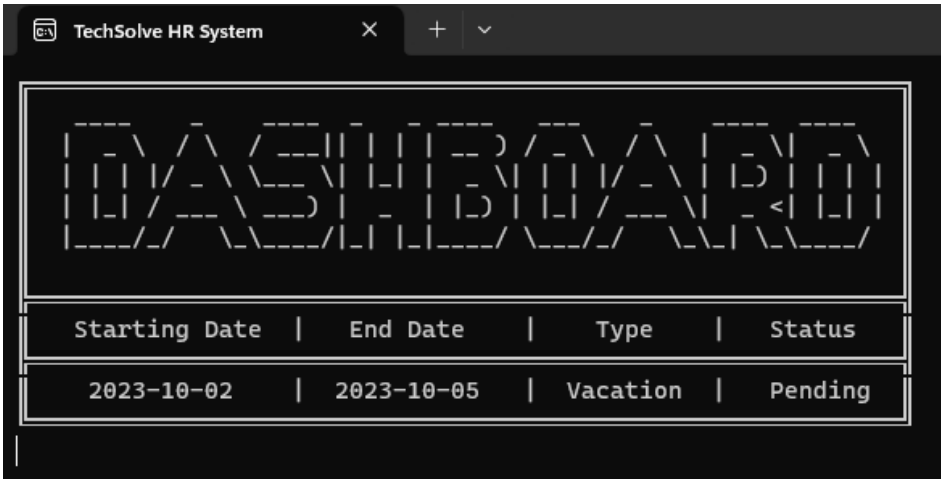


Figure 2.3 (Leave List) displays the starting date, end date, type, and status of the applied leave.

Access Company Resources

Figure 3.0

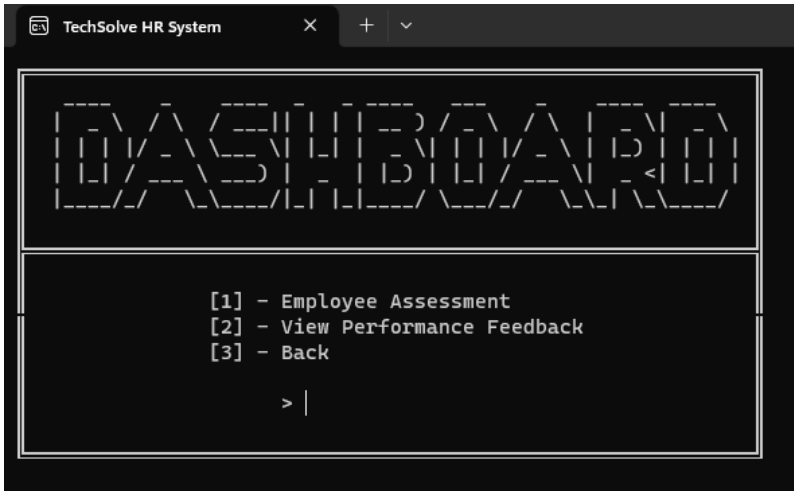


Figure 3.0 (Resources Menu) displays the the following options for the employee such as:

1. Employee Assessment - Gives admins the ability to choose an employee assigned to them by ID to evaluate.
2. View Performance Feedback - Gives the user access to performance feedback by their assigned manager.

Figure 3.1 **Figure 3.2**

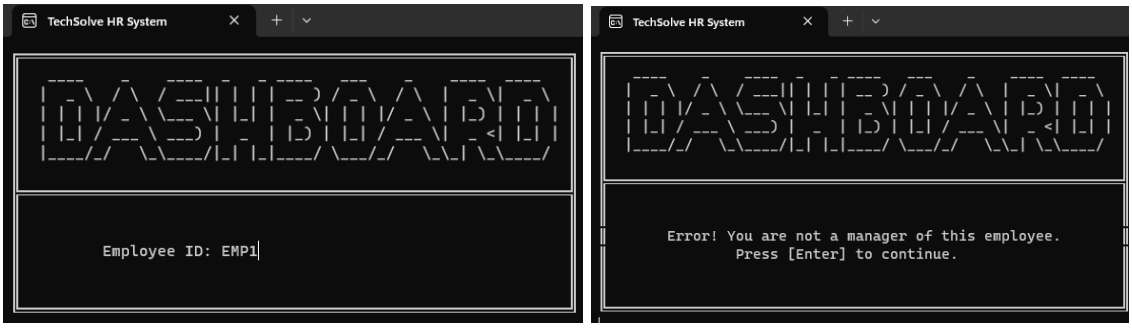


Figure 3.1: Requests the employee ID and checks if the user has admin privileges; if not, it displays an error message in Figure 3.2 stating, "You are not a manager of this employee."

Figure 3.3

Figure 3.3 presents a table featuring ratings from the employee's manager, including categories such as job knowledge, work quality, punctuality, productivity, communication skills, and an overall rating. Additionally, there is a section for comments provided by the manager.

Attendance Records

Figure 4.0

Figure 4.0 displays the menu of the attendance records feature, wherein it gives the user two options:

1. Add Record: Automatically registers the date and time from the user's time settings. It also provides manual input.
2. View Records: Displays a table that consists of the date, time in, and time out of the user.

Figure 4.1

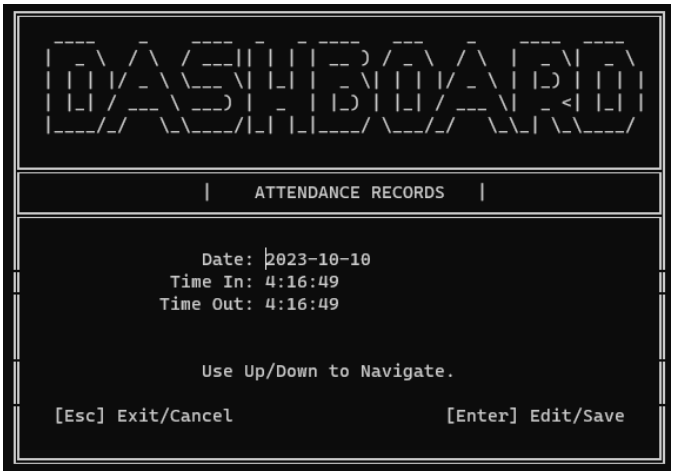


Figure 4.1 records the date and time from the user's system for attendance. Manual input can also be used to register attendance.

Figure 4.2

Figure 4.2 displays the registered attendance data

Overtime Records Menu

Figure 5.0

Figure 5.0 displays the menu options for the overtime recording feature

Figure 5.1

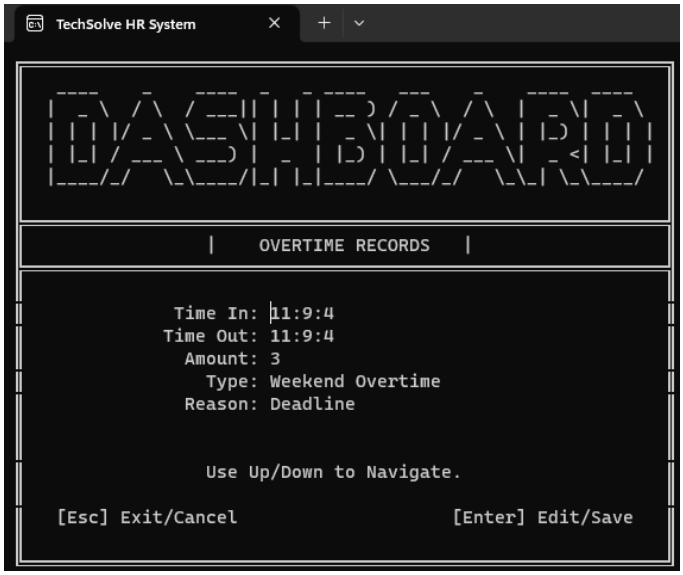


Figure 5.1 automatically registers system time, but can be changed by manual input as well. Asks for the amount of time added, the type of overtime, and reason.

Figure 5.2

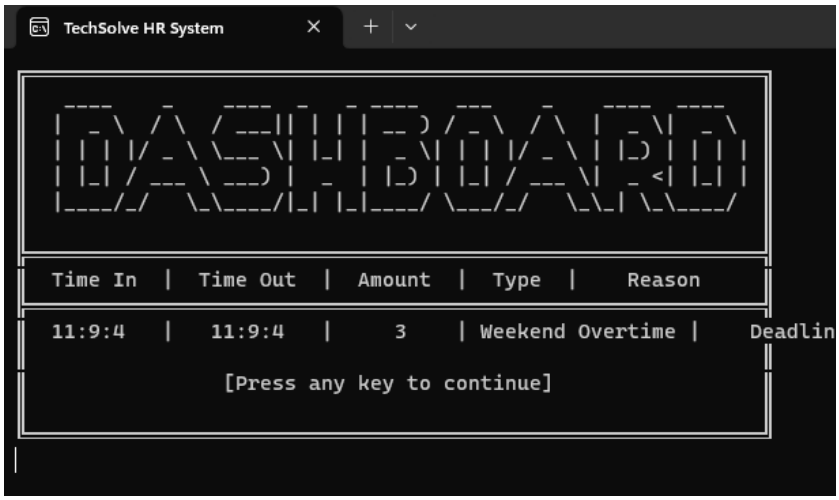


Figure 5.2 displays the registered overtime data. It shows the time in, time out, amount, type of overtime, as well a reason.

View Employees

Figure 6.0

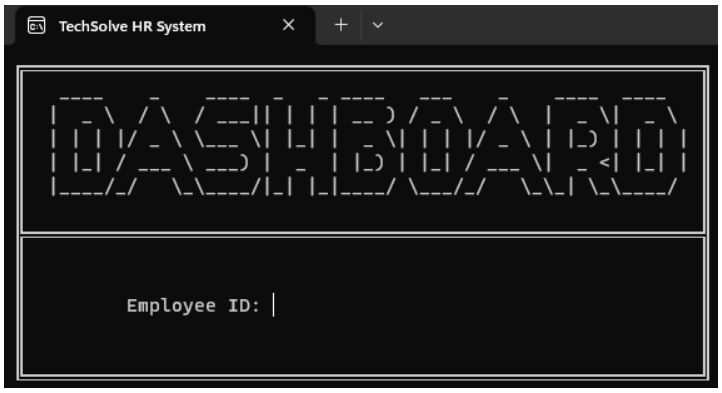


Figure 6.0 asks the user for an Employee's ID that they want to view. They are then able to traverse through the information of that employee (Figure 6.1, Figure 6.2).

Figure 6.1

Figure 6.2

Change Password

Figure 7.0

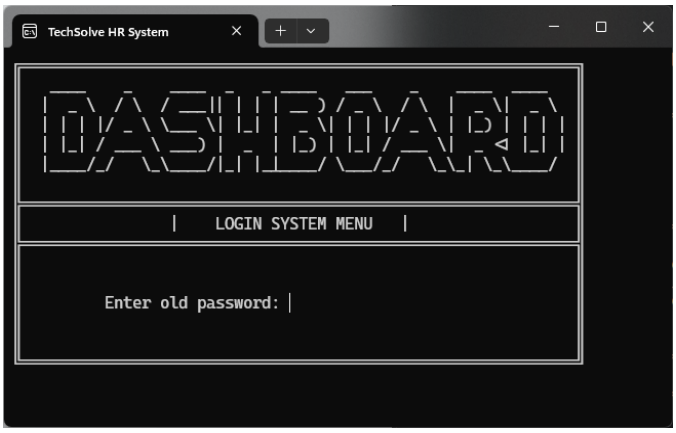


Figure 6.0 Prompts the user of their current/old password

Figure 7.

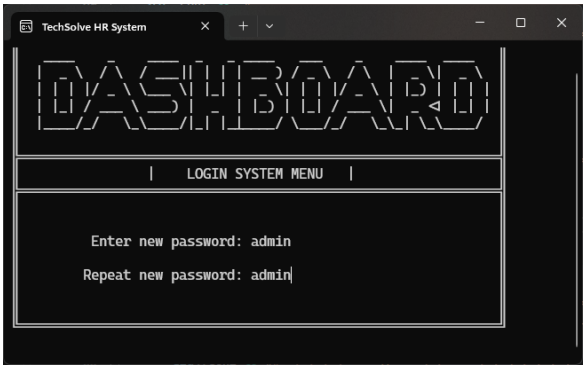
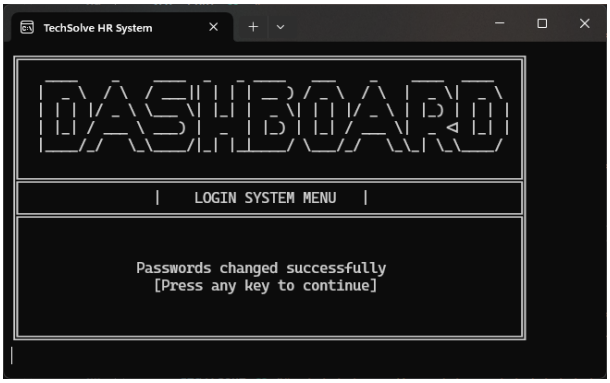


Figure 7.2



In Figure 6.1, the user is prompted to create a new password, but there are no constraints or specific requirements in place for the password's composition, such as minimum length, complexity, or restrictions on common words or patterns. Figure 6.2 displays a message that the password is changed.