

**Coursework (Group project )**

“Employee Management System”

**Module: -** [CST2550 Software Engineering Management and Development](https://mdx.mrooms.net/login/index.php?saml=on)

**Lab Tutor: -** Ahmed Eissa (Module Leader)

Adam Philpot

Olugbenga Oluwagbemi

**Group Members**(The Six-Coders System)

1.Chanchlani Akshar Hemantbhai (Leader)

2.Patel Abhi Jagdishkumar (Coder)

3.Galani Viren Sureshhai (Coder)

4.Jani Ayush Chiragkumar (Tester)

5.Mistry Yash(Tester)

6.Nirmala Suhas Manjunath(Tester)

**...INDEX...**

|  |  |  |
| --- | --- | --- |
| **Sr no.** | **Objective** | **Pg.no** |
| 1 | Introduction | 3 |
| 2 | UML design | 4 |
| 3 | Implementation | 4 |
| * Approach * About Make file * Why version control |
| 4 | Testing approach | 6 |
| * A statement of the approach * Details of test cases |
| 5 | Software demonstration | 7 |
| 6 | Conclusion | 9 |

**Executive Summary**

The Employee Management System (EMS), which aims to improve HR management with a user-friendly and flexible system, is presented in this research. The project's main goal was to create essential tools for managing employee data, namely the ability to add, edit, and find employee records. Reducing manual errors and increasing task efficiency were the objectives.

Preliminary results indicate that the EMS effectively simplifies HR functions and provides a strong basis for further improvements. The conclusion indicates a promising future, with initiatives to add new services and investigate untapped markets.

**Introduction**

Efficiency and organization are crucial in a busy work environment, which is where our Employee Management System (EMS) shines. This program is an easy-to-use application designed to assist companies of all sizes in managing and organizing their personnel data.  
  
Administrative tasks become less complicated and time-consuming with our EMS. HR managers will find the system ideal as it allows them to add employee profiles, update their data as the staff base expands, and remove departing employees from the system. It all comes down to managing personnel information without letting paperwork accumulate.

Additionally, locating specific information is fast. Our system's search function makes it easy to find someone's job title or arrange personnel by department. In addition to saving time, this frees up managers to work on more strategic assignments.  
  
Our EMS is unique in that it is very flexible. The system is prepared to grow with your business, giving you always access to the necessary resources. It's a comprehensive system built to assist the expansion of your business and optimize HR procedures; it's more than just a database.

By selecting our EMS, you're laying the groundwork for a workplace that is more orderly, efficient, and manageable in addition to investing in a dependable method of managing employee data. We promise to deliver you a system that will expand with you and meet your changing business demands for many years to come.

A black screen with white text

Description automatically generated

**Implementation of the UML Diagram**

By identifying the required classes and their relationships, we started the process of developing our EMS by following the blueprint that our UML design had supplied. We used a methodical approach, making sure that every class—such as Employee and Admin—had the proper features and techniques to allow our system to effectively manage HR-related duties in the real world.

The connections between these classes were then developed. To encapsulate further management features, for example, we implemented inheritance where Manager classes extended Employee classes. Associations were implemented to facilitate communication between various system components, such as data management and login sessions.  
  
Our UML diagram's structure was mirrored in the database schema, which offers a reliable backend for data storing and retrieval. In order to carry out the required tasks, we created functions and made sure they matched the use cases we had described.

Unit tests were built for individual components, and integration tests were used to examine how these components interacted with one another. A comprehensive testing regime was implemented to ensure that every aspect of the system operated as intended.  
  
We improved the system with constant feedback by iterating over our procedures, lining up our code with the UML design and making necessary adjustments to achieve our goals quickly and successfully.

**Development Environment and Collaboration Tools:**

Our team utilized an effective blend of tools and platforms to develop and coordinate on the Employee Management System (EMS).

* **Cygwin and Visual Studio Code:** By utilizing Visual Studio Code's extensive feature set and Cygwin, which allows Windows to have Unix-like functionality, we were able to construct an effective development environment. This combination allowed us to leverage more sophisticated programming tools and made it easier to run scripts and manage version control.
* **GitHub:** GitHub served as the host for our source code, offering a centralized repository for code reviews, distributed version control, and pipeline management for continuous integration and delivery.
* **Zoom and In-Person Meetings:** Our process required regular team meetings. In addition to meeting in person, we collaborated virtually using Zoom to better discuss and solve challenging issues.

**Coding And Data Management:**

We managed personnel data well by creating C++ classes that included the EMS's business logic and utilizing CSV files for durable data storage.

* **Make And Cmake Files:** Our build procedures were automated by these tools, which also took care of dependencies and made sure our application could be produced successfully in a variety of situations.
* **Catch2 Files:** We utilized Catch2 for testing, which offered a powerful framework for creating and running unit tests. This allowed us to consistently test our code, which helped us maintain good code quality.

**Project Management And Workflow:**

* **Azure Devops:** Using Agile approaches to manage activities and monitor progress, Azure DevOps helped us stay on schedule with our project.
* **Team Roles And Dynamics:** Our six-person team consisted of three testers who ensured the software was reliable and performing as intended, two developers who developed the system's functionality, and a Scrum Master who oversaw the project.

**Continues Collaborations:**

* Zoom was used for daily stand-ups and sprint retrospectives, while GitHub was used for code exchange and review among team members throughout the EMS's development. Our communication strategy was based mostly on these tools and in-person meetings, which helped us keep the team engaged and in sync while producing a high-caliber EMS.

**Version Control:**

* **Git:** used in conjunction with a platform that was integrated with Azure DevOps to handle code changes through version control. Collaboration, code reviews, and the continuous integration/continuous deployment (CI/CD) pipeline were made easier by this configuration.

**Testing Approach**

**Statement Of Testing Approach:**

We took great care in crafting our testing process to guarantee thorough coverage of all EMS features. We used the following two testing approaches:

1. **Unit Testing**: Catch2 was used to test each module's functionality separately. This made sure that before each part was included into the bigger system, it operated as intended.
2. **Integration Testing:** To make sure that various EMS components interacted with one another without any problems, we carried out integration tests following unit tests.

Because the testing process was iterative and followed the Agile development technique, the system could be continuously improved with every sprint.

**Details Of Test Cases:**

Test cases were created to address both edge cases and a range of anticipated application cases. These test cases are illustrative of their use:

* **Add Employee Test:** To test the 'Add Employee' feature, we entered some new employee information and made sure the database included the information

appropriately.

* **Test of Search Functionality:** A few searches were carried out to make sure that the functionality of the search could correctly retrieve employee data according to various parameters, such as name, ID, and department.
* **Data Integrity Test:** Tests were conducted to ensure that, while altering employee details, the database appropriately reflected the changes without damaging irrelevant data.
* **Error Handling Test**: We examined how the system would react in the event of erroneous inputs or actions, like trying to delete a worker who isn't actually there.

To guarantee consistency and repeatability in our testing processes, each test case contained preconditions, test steps, expected outcomes, and postconditions.

**Employment System Management Functionality**

**A black screen with white text

Description automatically generated**

**Administrative Access and Registration:**

Upon completing a secure registration process, an administrator (usually a manager) creates the company profile on the system. They are required to provide basic information including their name, the name of the company, the email address of the company, and a strong password that must be verified. To add even more security, a special security code is created.

**Employee Management:**

Admins can manage employee records using a number of functions after registering:

* **Add Employee:** Administrators may add new staff members by inputting their IDs and pertinent information, like name, email address, date of birth, department, position, and pay.
* **Edit Employee:** To make sure that employee data are always up to date and correct, administrators can edit existing employee details by looking them up using their unique ID.
* **Employee Search:** To make it easier to retrieve employee records, the system provides a simplified search option that allows employees to be located by department or ID.
* **Remove personnel:** This tool, which is essential for keeping current personnel data, enables administrators to take out an employee from the system by inputting their ID.
* **View All Employee Data:** The system offers a thorough overview of employee information at a look by displaying all employee data in a table style.

**Data Storage:**

All company and employee data is safely kept in CSV files, which are an easy-to-use yet efficient alternative to complicated database systems for managing data. This option promotes backup capability and accessibility.

**Class Structure and System Architecture:**

The EMS's backend logic is contained in clearly defined classes, each of which oversees managing a certain component of the system's operation, such as data processing or user authentication. The system is designed using best practices in object-oriented programming to guarantee data integrity and operational effectiveness.

**Conclusion**

Simplifying the management of workforce data has advanced significantly with the help of our employment management system. We've placed a strong emphasis on usability, security, and efficiency throughout development. Important HR tasks are centralized by the system, which makes it simple for administrators to add, update, and delete personnel as well as keep current records without the typical manual overhead.  
  
Important features include the ability to view all employee data in a tabular style and search employee records by ID or department highlight the system's ability to deliver quick and simple information access. Data portability and simplicity are ensured by storing data in CSV files.

In the future, as our user base expands, we hope to improve the user interface, add more features, and maybe switch to a more reliable database system. We will keep collecting input for next versions to make sure the EMS adapts to our users' evolving needs and the ever-growing field of personnel management.  
  
With its existing features and upcoming enhancements, the EMS is well-positioned to become as a crucial resource for companies looking for an effective and dependable HR management solution.