International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 06 | Jun 2026 www.irjet.net p-ISSN: 2395-0072

# Employee Salary Management System: Enhancements for Smart Cities

**Salem Hamdan**

[1]*College of Computing, Fahad Bin Sultan University, AIMD Lab, KSA, Tabuk*

---

**Abstract** - *This project presents a comprehensive and practical Java lab solution aligned with the structure required in soft engineering demands. It covers teamwork, Java GUI development, salary calculation processes, input validation, data presentation, and testing-based output preparation. Once the student adds the required names, IDs, and actual screenshots, the project will be ready for submission as a well-organized and professional course deliverable. It presents promising aspects for enhancements in terms of security, privacy, employing agent software technology to be effective for smart cities.*

**Key Words: Salary, system, method, agent, privacy, smart cities.**

## 1. INTRODUCTION

An Employee Salary Management System is a software application designed to help organizations manage employee salary information in an accurate, organized, and efficient manner. In many organizations, calculating salaries manually can lead to errors, delays, and difficulties in maintaining employee records. Therefore, using a computerized system helps simplify the process of storing employee details, calculating salaries, applying deductions or bonuses, and displaying final salary information clearly [1].

This system provides a practical solution for managing basic employee data such as employee name, ID, department, working hours, basic salary, allowances, deductions, and net salary. It can also support salary calculations based on predefined rules, reducing the chance of human error and improving the reliability of payroll operations. In addition, the system allows users to view and update salary records easily through a simple interface.

From a programming perspective, developing an Employee Salary Management System using Java helps students apply important concepts such as graphical user interface design, input validation, conditional statements, calculations, object-oriented programming, and data presentation. The project also provides an opportunity to understand how real-world business applications are designed and implemented [2].

Overall, the Employee Salary Management System is a useful and practical application that improves salary processing, enhances record management, and supports better decision-making within an organization.

In smart cities, an Employee Salary Management System plays an important role in supporting efficient, transparent, and data-driven administration. Smart cities depend on digital systems to manage services, institutions, and human resources effectively. By automating salary calculation, employee record management, allowances, deductions, and payroll reports, this system reduces manual errors and saves time for both employees and administrators. It also supports faster decision-making by providing accurate salary data and organized records. In addition, when connected with other smart city platforms, such as attendance systems, human resource systems, and financial management systems, it can help improve institutional productivity, accountability, and service quality. Therefore, an Employee Salary Management System is an essential digital tool that contributes to the modernization and efficiency of smart city organizations [3].

When an Employee Salary Management System is enhanced with advanced technologies such as security mechanisms, big data processing, deep learning, privacy protection, and agent-based software, it becomes more powerful, reliable, and suitable for modern smart city environments. Strong security features protect salary records from unauthorized access, data theft, and manipulation, while privacy protection ensures that sensitive employee information, such as salaries, deductions, and personal details, is handled safely and confidentially. Big data techniques allow the system to manage large amounts of employee records across different departments or institutions, making payroll analysis and reporting faster and more accurate. Deep learning can support intelligent prediction and decision-making, such as detecting unusual salary transactions, predicting payroll costs, and identifying errors or fraud patterns. In addition, agent-based software technology can automate tasks such as collecting attendance data, updating salary records, generating reports, and communicating with other systems. As a result, the enhanced system improves efficiency, accuracy, transparency, and trust in salary management within smart city organizations [4].

If these enhancements are ignored in smart cities, an Employee Salary Management System may become weak, unsafe, and unable to meet the needs of modern organizations. Without strong security and privacy protection, sensitive employee data such as salaries, personal information, deductions, and bank details may be exposed to unauthorized access, misuse, or cyber-attacks. Without big data capabilities, the system may fail to handle large numbers of employee records across different smart city insti-

tutions, leading to delays, inaccurate reports, and poor decision-making. Ignoring deep learning may also reduce the system's ability to detect abnormal salary transactions, payroll fraud, or repeated calculation errors. In addition, without agent-based software technology, many processes may remain manual, slow, and difficult to coordinate with other smart city systems. As a result, the organization may face financial losses, low employee trust, administrative inefficiency, and serious privacy risks.

## 2. SIMPLE OVERVIEW ABOUT THE PROJECT

The project solves a practical administrative problem. In many small offices, salary information is handled manually using paper forms or basic spreadsheets, which increases the probability of calculation mistakes and duplicate work. This application provides a cleaner interface, validates user input, and stores the added employees inside a table so that the user can review the results immediately.

The implemented system contains a graphical user interface, salary processing logic, summary statistics, sample data loading, and row deletion. These features demonstrate the use of Java GUI programming, object-oriented design, arrays or collections, conditional logic, event-driven programming, and data presentation in tables.

## 3. DESCRIPTION OF BENEFIT OF THE PROJECT (FEASIBILITY)

The project is technically feasible because it is implemented using standard Java libraries that are available in a typical educational environment. No external paid software or database server is required. The system can be compiled and executed through any online Java compiler that supports Swing or through a desktop IDE such as IntelliJ IDEA, Eclipse, or NetBeans.

The project is economically feasible because the development cost is very low. It mainly requires students' programming effort and a normal computer. For a small company or laboratory setting, the system can reduce manual salary calculation time and decrease human errors, which provides clear value without additional infrastructure cost.

The project is operationally feasible because the interface is simple and understandable. A staff member only needs to type four input values and click the add button. The result appears directly in the table, and the overall salary statistics are updated automatically. This makes the system suitable for demonstration during the project presentation and easy for non-technical users to understand.

The project is academically feasible as well because it maps directly to course learning outcomes: team collaboration, Java coding, testing, and documentation. It is a realistic lab-scale problem that still demonstrates several important programming concepts in one integrated solution

## 4. EXPLAIN THE MAIN PIECES OF CODES

The program is organized around a main class called Employee Salary Management System that extends JFrame. This class creates the application window and all interface components such as text fields, labels, buttons, a combo box for position selection, a table model for displaying employees, and a summary label for live statistics.

The Employee inner class is responsible for representing one employee object. It stores the employee name, basic salary, allowance, position, and calculated bonus. A dedicated method computes the total salary by summing the basic salary, allowance, and bonus. This is a good example of encapsulating related data and behavior inside one object.

The add Employee method is one of the most important code sections. It reads the user input, checks whether fields are empty, converts salary values from text to numbers, prevents negative entries, calculates the bonus according to the selected position, and then adds the result to both the internal employee list and the visible JTable. This method also updates the summary and clears the form after successful insertion.

The calculate Bonus method implements the business rule of the project. Different positions receive different bonus percentages. This part demonstrates the use of conditional branching through a switch expression. It also makes the program easier to extend later if the company adds new job categories or bonus policies.

The update Summary method computes the grand total of all salaries and the average salary for the currently listed employees. This gives the program additional analytical value beyond only storing rows. The user therefore obtains immediate managerial information from the same interface.

The load Sample Data method was added to simplify project presentation. During the live demo, students can populate the system quickly with predefined employees and show how the table and salary summary change in real time. This improves demonstration quality and reduces the risk of typing errors during presentation.

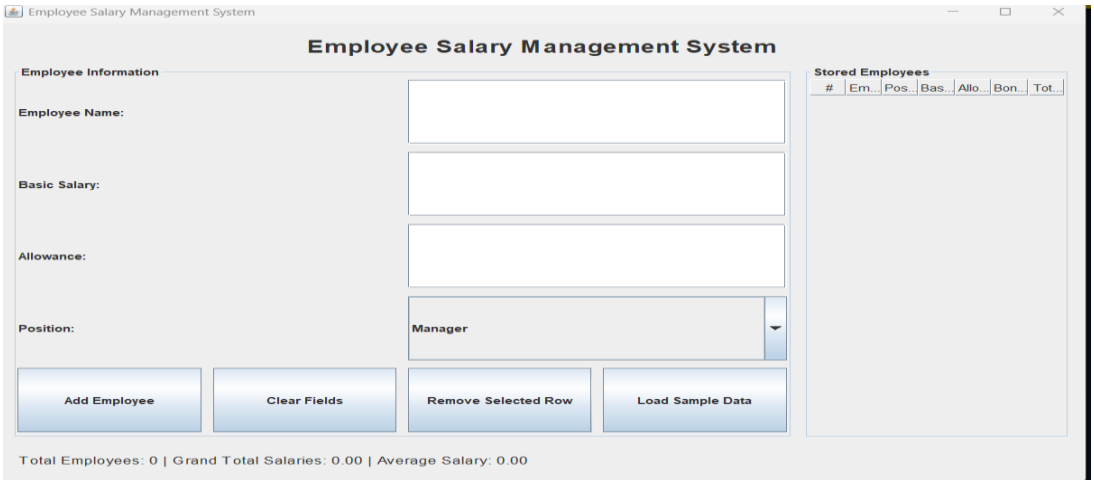
## 5. OUTPUT OF THE PROJECT

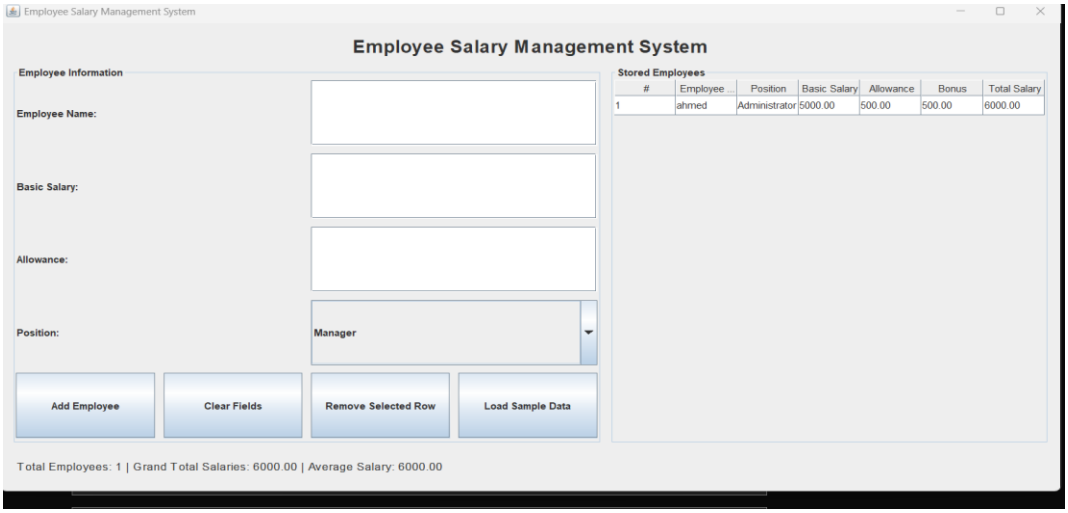
When the program starts, the user sees a clean desktop window with input controls on the left side and a data table on the right side. The title clearly identifies the system as an Employee Salary Management System.

After entering an employee name, basic salary, allowance, and position, the user clicks Add Employee. The application calculates the bonus percentage based on the position and inserts a new row into the table. Each row shows the employee number, name, position, basic salary, allowance, bonus, and total salary.

The summary label at the bottom of the screen displays three important outputs: the total number of employees, the grand total of all salaries, and the average salary. If a row is selected and removed, the system recalculates the summary values automatically. This proves that the system is dynamic and responsive.

The following spaces are intentionally prepared so the student can paste real screenshots from execution before final submission. The following figures illustrates the interfaces of the system.



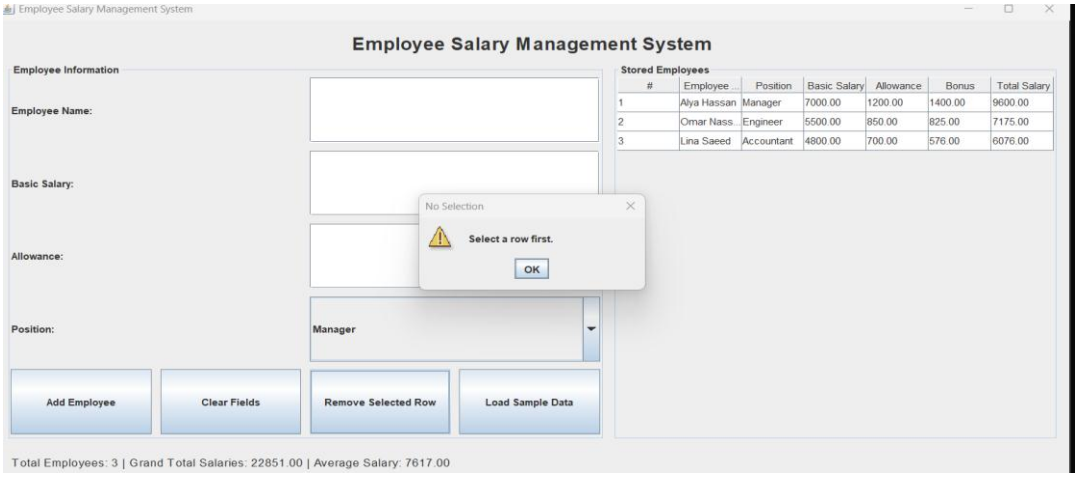



**Fig-1: Interfaces.**

## 6. CHALLENGES LINKED WITH PROMESSING ENHANCEMENTS

There are many challenges related to this system when integrating with smart cities. In smart cities, the sensory data are considered as big data, where special data mining methods have to be employed to end functionalities of the system [5]. In addition, most of systems are supported with AI and deep learning techniques. In this context, when integrating the system into medical systems, such as [6, 7, 8], many functionalities have to be trained and tested according to the basic steps of AI domain. Moreover, security of this system can be enhanced to be of high level of reliability in smart cities, where security of websites must be considered in the system will be available for online-usage [9]. Furthermore, the system can be enhanced using location based services [10]. In this context, privacy issue must be ensured because users can search for the nearest companies the user wants to explore based on his\her real location. Actually, many works addressed the privacy issue in location based services, such as [11, 12, 13, 14, 15]. Beyond privacy protection issue, there is another security issue related to enhance the system based on agent software technology. Indeed, agents are common in smart cities to end daily activities. Managing salary of employees in terms of collecting taxes and other financial aspects requires using mobile agent, which is a mobile code that can travel through all functionalities of the system to collect the desired reports. In this context, security of agents are critical due to attacks that may be applied by destination machine or by man in the middle attackers. Many works addressed security of agents, such as [16, 17, 18].

## 7. FUTURE DIRECTIONS

In the future, the project can be enhanced by connecting the application to a database such as MySQL so employee records remain saved after the program closes. This would make the project more realistic for practical business use.

Another improvement would be generating salary reports in PDF or Excel format. Such reports would help HR or accounting staff share salary summaries with management more efficiently.

The system can also be extended with user authentication, search functionality, update functionality for existing records, tax deduction calculations, and department-based filtering. These extensions would transform the current lab project into a more complete payroll management solution.

## 8. CONCLUSION

This project provides a complete and realistic Java lab solution that satisfies the structure requested in the uploaded course template. It demonstrates teamwork, Java GUI programming, salary calculation logic, validation, data display, and testing-oriented output preparation. After the student fills in names, IDs, and real screenshots, the project can be submitted as a polished course deliverable.

## ACKNOWLEDGEMENT

The author take help of ChatGPT to present illustrative figures as well as for proof reading. However, the basic scientific content is presented by the author.

## REFERENCES

[1] Hao, Zhang, et al. "Salary management system for small and medium-sized enterprises." Physics Procedia 24 (2012): 2255-2260.

[2] Wang, Tianqi, Ning Li, and Houran Li. "Design and development of human resource management computer system for enterprise employees." Plos one 16.12 (2021): e0261594.

[3] Kaluarachchi, Yamuna. "Implementing data-driven smart city applications for future cities." Smart Cities 5.2 (2022): 455-474.

[4] Berendt, Bettina. "AI for the Common Good?! Pitfalls, challenges, and ethics pen-testing." arXiv preprint arXiv:1810.12847 (2018).

[5] Mohamad Shady Alrahhal and Adnan Abi Sen, "Data mining, big data, and artificial intelligence: An overview, challenges, and research questions," 2018.

[6] Mona Alfifi, Mohamad Shady Alrahhal, Samir Bataineh and Mohammad Mezher, "Enhanced Artificial Intelligence System for Diagnosing and Predicting Breast Cancer using Deep Learning," International Journal of Advanced Computer Science and Applications, vol. 11, no. 7, 2020.

[7] Mohamad Shady Alrahhal and Eftkhar Alqhtani, "Deep learning-based system for detection of lung cancer using fusion of features," International Journal of Computer Science and Mobile Computing, vol. 10, no. 2, pp. 57-67, 2021.

[8] Mohamad Shady Alrahhal and Majed Abdullah Albarrk, "A Survey of the COVID-19 Epidemic Through the Eyes of Artificial Intelligence and Deep Learning: Challenges and Research Questions," 2020.

[9] Majed Abdullah Albarrk and Mohamad Shady Alrahhal, "Web Applications Security: More Collaboration," 2020.

[10] Mohamad Shady Alrahhal, Maher Khemakhem and Kamal Jambi, "A Survey on Privacy of Location-Based Services: Classification, Inference Attacks, and Challenges," Journal of Theoretical and Applied Information Technology, vol. 95, no. 24, 2017.

[11] Mohamad Shady Alrahhal, Maher Khemakhem and Kamal Jambi, "Agent-Based System for Efficient kNN Query Processing with Comprehensive Privacy Protection," International Journal of Advanced Computer Science and Applications, vol. 9, no. 1, 2018.

[12] Mohamad Shady Alrahhal, Muhammad Usman Ashraf, Adnan Abesen and Sabah Arif, "AES-Route Server Model for Location Based Services in Road Networks," International Journal of Advanced Computer Science and Applications, vol. 8, no. 8, 2017.

[13] Hosam Alrahhal et al., "A symbiotic relationship based leader approach for privacy protection in location based services," ISPRS International Journal of Geo-Information, vol. 9, no. 6, p. 408, 2020.

[14] Abdullah S. Alyousef, Karthik Srinivasan, Mohamad Shady Alrahhal, Majdah Alshammari and Mousa Al-Akhras, "Preserving Location Privacy in the IoT against Advanced Attacks using Deep Learning," International Journal of Advanced Computer Science and Applications, vol. 13, no. 1, 2022.

[15] Mohamad Shady Alrahhal, Maher Khemekhem and Kamal Jambi, "Achieving load balancing between privacy protection level and power consumption in location based services," International Research Journal of Engineering and Technology, vol. 5, no. 3, pp. 619-625, 2018.

[16] Bandar Alluhaybi, Mohamad Shady Alrahhal, Ahmed Alzhrani and Vijey Thayananthan, "A Survey: Agent-based Software Technology Under the Eyes of Cyber Security, Security Controls, Attacks and Challenges," International Journal of Advanced Computer Science and Applications, vol. 10, no. 8, 2019.

[17] Bandar Alluhaybi, Mohamad Shady Alrahhal, Ahmed Alzhrani and Vijey Thayananthan, "Dummy-based approach for protecting mobile agents against malicious destination machines," IEEE Access, vol. 8, pp. 129320-129337, 2020.

[18] Bandar Alluhaybi, Mohamad Shady Alrahhal, Ahmed Alzhrani and Vijey Thayananthan, "Achieving self-protection and self-communication features for security of agent-based systems," 2020

## How to Run the Project

- Install JDK 17 or any recent JDK version that supports Swing.
- Save the source file as EmployeeSalaryManagementSystem.java.
- Compile the file using: javac EmployeeSalaryManagementSystem.java
- Run the program using: java EmployeeSalaryManagementSystem
- For online execution, use a Java environment that supports graphical Swing applications; otherwise, run it locally in an IDE.

## APPENDIX

This appendix provides some details related to implementation stage and how to run the system.

| Component / Method | Purpose in the Program |
|---|---|
| JTextField and JComboBox | Collect employee data from the user. |
| DefaultTableModel and JTable | Display all inserted employees in tabular format. |
| addEmployee() | Validate inputs, calculate bonus, create employee object, and append a new table row. |
| calculateBonus() | Apply salary policy based on job position. |
| updateSummary() | Compute total employees, grand total salaries, and average salary. |
| loadSampleData() | Insert demo records quickly for live presentation. |
| removeSelectedEmployee() | Delete one selected record and refresh numbering and summary. |