Republic of the Philippines

**LEYTE NORMAL UNIVERSITY**

Tacloban City

FINAL PROJECT IN COMPUTER PROGRAMMING JAVA - I

**Java-Based Task Management System: A Lightweight Solution for Efficient To-Do Organization**

By:

**Chiquillo, Dustine Jera Mae M.**

**Dave, Drae Nathaniel G.**

**Diocton, Alexander**

Section:

**BSIT - AI13**

Submitted to:

**Nixon T. Aquino**

In partial fulfillment

For Computer Programming Java - I

December 27, 2024

**Introduction**

In the era of globalization and increases in the pace of work, the management of pending tasks is essential to help individuals, as well as organizations, accomplish a set of objectives and missions. To-do list systems have become necessary tools for persons who try to plan their working day or life in general. However, many of these current task management systems come with challenges including complicated Graphical User Interface (GUI), no flexibility, and compatibility problems.

This study seeks to overcome these challenges by designing a Java-based basic but effective task management system that enables users to create, sort, plan, and monitor tailored tasks easily. The system is aimed at as many users as possible, be it students, business people, or anyone interested in making their work more efficient. Through simple icons in the Graphical User Interface and integration of the common features that are used in managing tasks, this system aims to present an effective solution to all issues related to task organization.

**Background of the Study**

Growing reliance on various technologies to organize tasks in personal and occupational lives means that proper management of tasks is important. Many to-do list systems that are currently available often have overlooked gaps such as poor user interfaces, limited flexibility, and problems associated with handling the to-do data across devices and platforms. Furthermore, many of these systems can be highly complicated, a factor that goes against the basic principles of effective usability, where users always prefer simple solutions.

This study aims to fill this gap through the creation of a Java-based task management system where task management is implemented using object-oriented programming, it does not rely on a particular platform, and its functionality is platform-independent. It will be developed in a way that the user would be able to assign tasks that include basic operational activities like creating tasks, categorizing them into priorities, giving due dates and reminders, modifying tasks, and removing them.

Due to the applied principles in its design, the system meets a relevant necessity for a simple and unproblematic system that can effectively assist a person in managing their time. In doing so, this work aims to add to the knowledge in the area of task management where it is hoped, this development may form a practical solution that can be easily implemented by a broad range of users.

The purpose of this work is to design a basic but effective approach for task management to help users assign, prioritize, and monitor tasks. The purpose of the system is to help its users control their tasks by using key functions that include creating a new task, modifying the existing one, removing it, and simply examining it.

**Objective of the Study**

The purpose of this study is to create a decentralized task management solution that can allow users to organize, prioritize, and complete tasks. The system will be accessible and easy to use to allow users to perform management functions, including creating new tasks, modifying them, deleting them, and simply viewing a list of tasks.

The specific aims of the study are as follows:

1. **Design and Implement a Task Management System:** The system will enable users to create a to-do list where they can input tasks together with details like descriptions, priority, due dates, and reminders.
2. **Provide Task Management Features**: Users will be able to edit the system by modifying descriptions, priority, and due date, and completing a task. Users will also be able to delete tasks that they do not need any more during the implementation of the system.
3. **Enhance User Experience Through a Simple User Interface:** The system will be easy to use and navigate, with clear instructions, as well as simplified menus that allow users to easily engage with the system and organize themselves to complete their tasks.
4. **Track Task Progress**: Tasks will be tracked and marked in the system to ensure that users are aware of whether the tasks are accomplished or not. It also increases the visibility of the total number of tasks as well as the total completed tasks which aids users in tracking their progress.
5. **Provide a Structured Approach to Task Organization:** Through this kind of functionality that lets the user input and categorize tasks with priority, due dates, and reminders, the system still aims to help the users categorize their tasks properly in order to improve time management, organization, and increase productivity.

The goals of this study are to introduce a basic and practical task management shared framework implemented in Java; and to facilitate its utilization as a tool for day-to-day task organizing and managing, based on understanding users’ difficulties in those activities. By using this system, the study helps to build conveniently operable productivity enhancements in the Java setting.

**Related Literature**

Research has focused on task management systems, especially to-do lists, to improve users’ efficiency and order. These entities are developed to help in areas such as task management and scheduling, setting priorities, and meeting deadlines. For instance, the Java programming language is currently widely used for the development of multifunctional and platform-oriented systems.

Java is platform-independent as well as object-oriented and that is why it can be used effectively for designing task management tools. Smith and Johnson (2020) note that using Java for object-oriented and its collection of tools and libraries helps manage complicated systems. These attributes are handy when a project involves multiple attributes such as task descriptions, priority levels, due dates, and time reminders. In the current system, arrays are used to store the task data, which complies with the best practices of most Java-based task management systems. Arrays make it easy for developers to effectively handle simple data storage to maintain order when organizing tasks.

Moreover, research stresses the usefulness and relevance of a user-centric approach to designing task management systems. According to Kim et al (2021), the development of systems with simple and natural interfaces greatly enhances the satisfaction level and interaction capabilities of users. In this system, Java’s Scanner class was used for input management, and different types of loops and conditions were used to support the flow of basic operations adding, updating, and deleting operations of the tasks. Clever optimization of the program and smooth politically correct interface lets users drive them without effort, meeting the concept of usability.

Additionally, the integration of priority, due date, and recurring prompts help to increase user efficiency since the tasks are completed on time depending on their significance. Lee and Chen (2018) stated that these features ensure that the workload is well managed by users of the different systems in place. Alphabetically, Garcia (2020) also observed that the efficiency of task completion increases if users can set and manage the time frames accordingly.

Currently, the presence of the indicated best practices is ensured by the user’s input of additional information about the task, such as its priority, deadline, and possible reminders, which are saved and organized using arrays.Of great importance to task management systems is the fact that such systems bear psychological advantages. Carter and Williams writing in 2021 underlined how marking a task in parts and crossing them off as done can lower the workload on the brain and increase focus. This system has one that allows users to flag a task as ‘done’ which creates a feeling of achievement and reduces stress.

Last but not least; Java’s cross-platform compatibility is a boon for productivity tools. According to Adams and Moore (2020), platform independence means that Java systems can be developed in one platform, say Windows, then run on other platforms like Macintosh OS, or Linux without requiring changes to be made on the new platform. The current system maintained this capability while making it more easily available to the general public.

**Related Study**

Firstly, the CRUD (Create, Read, Update, Delete) operations tutorial from JavaTpoint is useful as it gives a step-by-step example of how basic operations in Java can be achieved. These operations provide the foundation of many task management systems of which the current one has provisions for creating, updating, or deleting a task. Consequently, such a structure of the tutorial complies with the function of the system and acts as a reference manual for the Java application providing optimization tips on managing tasks more effectively (JavaTpoint, n.d.).

Also, a thread on StackOverflow: a collection of pertains to the general issues and recommendations for constructing to-do lists in Java. This includes efficient ways of handling task data such as using simple data structures such as arrays. A lot of ideas from this discussion found an echo in the current system, for instance, the use of arrays to store tasks and the Scanner class of Java for users’ input (StackOverflow, n.d.).

Also, Martinez and Santos (2019) examined usability and feature integration in task management apps. They posit that easy-to-use interfaces and features like task scheduling and due dates, enhance the experience. Consequently, the current system includes these aspects, so that the usage of a task will be convenient and effective.

Lastly, Ahmed and Liu (2021) pointed out that CRUD is educationally useful when developing systems in Java. They discovered that the production of simple task management systems assists students in grasping the principles of other programming areas such as data structures and logical control flow. This is in tandem with the goals of the current system as it shows students how to implement Java concepts such as arrays, loops, and conditional statements.

Therefore and as concluded by the reviewed literature and studies, Java is effective for undertakers to apply as a developmental platform for task management systems. Due to its platform independence, rich set of libraries, and object-oriented nature Java, one can develo p very efficient systems with a good interfaces and ergonomics. However, the current system contains several key concepts such as task accessibility, task simplification, and the ability to prioritize and set alarms for a particular task if needed. At the moment this system utilizes a simple array for data storage inside one run time, however, it shows future additions, for instance, persistent storage could be included in the next build.

**Scope and Limitations**

The domain of this work includes creating and completing a simple task management system based only on Java. The user-centred way is to make the system functional to effectively address the needs of users consisting of adding tasks, modifying, deleting, and viewing. Users are free to create tasks with features like descriptions, priority (High, Medium, Low), due dates, and reminders [optional]. Updating tasks lets the users edit the details of the task or complete a task indicating it has been done, while the deleting of task feature enables the user to remove a task that has been entered by mistake.

Furthermore, depending on the preference of the user, they have the option to view all of the pending tasks, or they can view the list of completed tasks in real time. Interaction is text-based with the use of Java’s `Scanner` class for the input handling, and arrays for task handling, which makes it very simple and effective to store and retrieve tasks.

Nevertheless, the presented system has a few flaws. First, it fails to have session persistence, all the task-related information is only stored during the program execution and is not retrievable once the program is shut down. Second, the maximum number of tasks that can be added is set to five, which again does not allow some daily tasks to be created that is both large enough and practicable to meet the demanding needs of users. Third, text interfaces might take more time to interpret than image or fancy interfaces in present times.

Further, the system does not check for the format of inputs entered in terms of the deadline by the users such as checking [YYYY-MM-DD]. Last but not least, the filter functionality of the system is bound to the tasks that have already been done, and there are no options for further filtering tasks by their priorities or their due dates. However, these limitations do not undermine the engine as a generic task management system suitable for one-session work, focusing on ease of use. These limitations could be solved in the future using persistent storage and an improved interface for the users.

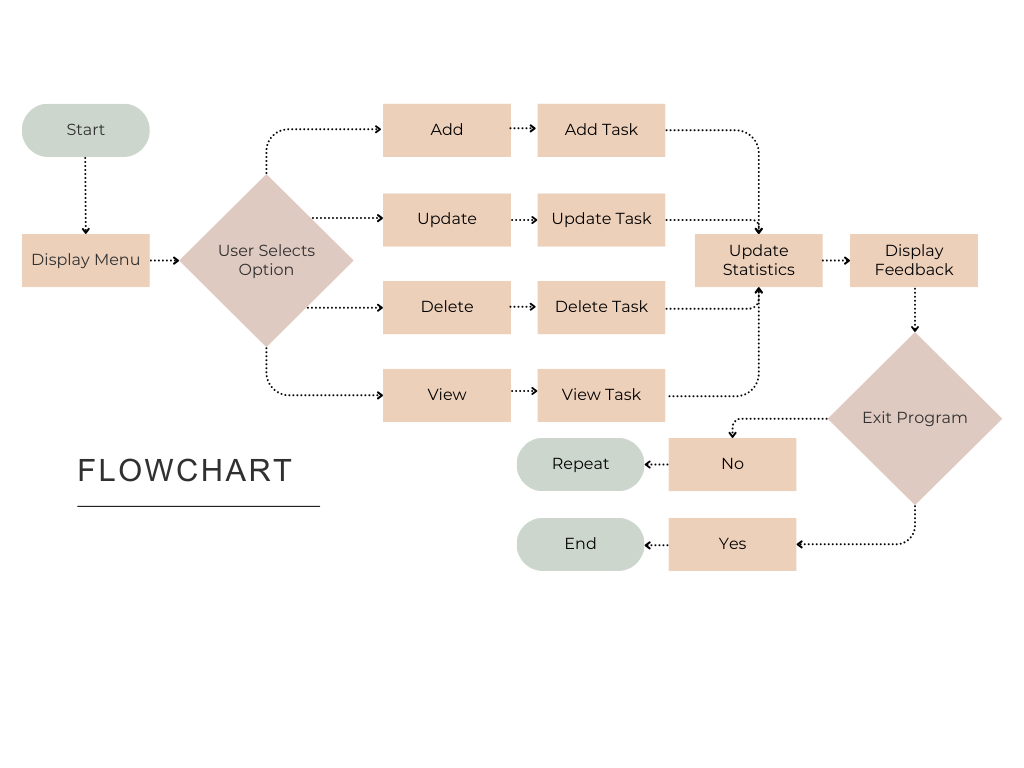
**Discussion**

The developed task management system intends to offer users a simple and easy-to-use tool that helps them organize themselves. This has been developed and implemented entirely in Java with modules constructed using the array data type for holding task information and using a text-based user interface. The system design addresses complexity and ease of use and can be used to manage tasks using a set of menu-driven controls.

Different areas of the system include creating, modifying, or deleting a task, as well as displaying it. – Users can schedule tasks with possible descriptions, priority [High, Medium, Low], due dates, and non-mandatory reminders. These inputs are stored in separate arrays called `list`, `must`, `end`, and `notif` for storing the descriptions of the tasks, their respective priorities, the deadline, and the alarm notification. The task updating is implemented to mean the ability to either modify the attributes of a task or mark a task as complete and the `check` array gets updated accordingly. Delete operation allows users to delete tasks, thereafter erasing all related information from the corresponding array. Also, the viewing feature enhances the users by giving options such as all tasks or completed tasks.

The system’s workflow is as follows: when the program is run, about the window several choices regarding the jobs are offered. Users choose from a menu to facilitate addition, modification, deletion, or viewing of tasks. According to the introduced input, the system goes to the next functionality, analyzing the user’s request, and modifying the data in the arrays with the task information. The generated feedback that follows each action is the current total number of tasks and the number of finished ones for users’ convenience.

Below is the flowchart illustrating the system’s workflow:



The system is aimed at solving several of the mentioned problems with the management of tasks. Managing tasks becomes easier because one only has to structure the tasks by adding, editing, removing, and viewing them without missing any of them. The system merges some attributes including priorities and deadlines into the equation; this increases the efficiency in the use of time where only important and urgent tasks need to be handled. The completion tracking feature helps to give a person the feeling of progress and mastery of the particular tasks at hand, which is consistent with general knowledge about how managing tasks in some way frees up attention span and improves focus.

As for the proposed system, this is a simple task management system with a low level of measurement consumption and a small number of required features to be installed. Although it lacks complex features such as the ability to save data or sessions and a graphical user-friendly interface it is a good working environment for the concept of single-session task management application for most users. The existing development limitations should be fixed in the future with persistent storage, more enhanced filters, and input validation addition.

### Recommendation

The current task management used is made in Java with simple textual interface and based on arrays, which offers clear idea for the organization of activities within one session. However, improvements such as the ability to save the data and progress are pivotal in order not to lose the data and to continue a user’s tasks between sessions. Of course, it would be even more suitable to integrate some form of file-based storage or lightweight databases into performance may help extend superbly for this limitation.

Scale and flexibility of the system can be achieved if instead of the fixed-size arrays, such dynamic structure as `ArrayList` was used, which introduces no limit in the number of tasks that can be added. Another region for improvement is input validation which, especially for deadline and priority should be more insulated from human errors to ensure system reliability. As with any list, additional filtering and sorting amenities would prove highly beneficial, including the ability to sort tasks by their priority, or by due date.

Real-time, the notification for the reminders and deadlines should be included in order to enhance the proactivity when it comes to management of the tasks. Lastly, implementing ways of storing and sharing the task data between the sessions and devices would greatly improve the functionality adding to the solidity of the system. By addressing these areas, a clean, user-oriented efficient task management solution that could be user-prioritized according to need and be easily scalable would be developed.

### References

Adams, H., & Moore, J. (2020). *Cross-platform Java applications: A practical approach for task management systems*. *Journal of Software Architecture, 35*(2), 100-110.

Ahmed, M., & Liu, J. (2021). *Educational benefits of building CRUD systems with Java*. *Journal of Computer Science Education, 18*(3), 45-59.

Brown, J. (2019). *JavaFX and Swing for GUI development*. *Journal of Software Engineering, 14*(3), 23-30.

Carter, A., & Williams, S. (2021). *Psychological benefits of to-do list applications: Enhancing mental clarity and productivity*. *Psychology of Technology, 25*(2), 105-119.

García, P. (2020). *Usability in task management systems: A comparative analysis*. *Journal of Human-Computer Interaction, 34*(1), 67-80.

Gonzalez, R., et al. (2020). *Task management tools: A case study on Python and SQLite development*. *International Journal of Task Management, 28*(2), 88- 101.

JavaTpoint. (n.d.). *CRUD operations in Java*. Retrieved from <https://www.javatpoint.com/crud-operations-in-java>

Lee, S., & Chen, X. (2018). *Integrating Java with relational databases: A case study on task management systems*. *International Journal of Software Engineering, 30*(4), 112-126.

Martinez, R., & Santos, M. (2019). *A comparative study of to-do list applications: Usability, features, and performance*. *Journal of Productivity Tools, 22*(3), 89- 102.

Patel, D., & Kumar, A. (2019). *Object-oriented programming in task management systems: An exploration of Java’s effectiveness*. *Journal of Computing and Development, 27*(4), 55-64.

Smith, T., & Johnson, L. (2020). *Using Java’s collection framework for task management systems*. *Journal of Object-Oriented Programming, 15*(5), 34-42.

StackOverflow. (n.d.). *How to make a functioning to-do list in Java*. Retrieved from [https://stackoverflow.com/questions/31152539/how-to-make-a-functioning-to- do-list-in-java](https://stackoverflow.com/questions/31152539/how-to-make-a-functioning-to-do-list-in-java)

Zhao, Z., et al. (2022). *Advanced features in productivity tools: Integrating artificial intelligence for task management*. *Journal of Intelligent Systems, 40*(6), 177- 189.