**Efficient Task Management: A C++ To-Do List Application**

Abstract

This research presents a to-do list application developed using Qt Project and Visual Studio, employing a Model-View-Controller architecture. The application facilitates efficient task management through intuitive user interfaces and robust backend functionalities. Key features include task creation, deletion, modification, and deadline management, leveraging Qt's cross-platform capabilities for deployment. The project highlights advancements in user interface design and software architecture, providing a scalable solution for personal and professional task organization. The implications of this work extend to enhancing productivity and time management practices through intuitive software tools.

# I. Introduction (Heading 1)

Abstract Task management is an essential component of productivity in both personal and professional settings. Many individuals struggle with keeping track of their tasks, leading to inefficiencies and missed deadlines. This project presents a C++-based to-do list application designed to help users efficiently organize their tasks. The application allows users to create, categorize, prioritize, and update tasks dynamically. By implementing object-oriented programming (OOP) principles, the system ensures modularity and scalability. This paper discusses the system's architecture, key features, and benefits, highlighting its role in enhancing task management.

Introduction Task management is a fundamental aspect of maintaining efficiency in personal and professional life. As responsibilities and commitments increase, individuals often find it challenging to track their tasks effectively. A lack of an organized system can result in missed deadlines, decreased productivity, and increased stress. The need for an intuitive and efficient tool to manage tasks is more critical than ever.

This project introduces a to-do list application developed using C++, designed to provide users with a streamlined method for managing their daily tasks. The primary goal of this project is to offer users an efficient system that allows them to organize their tasks, set priorities, and keep track of pending and completed activities.

The application features a structured approach to task organization, allowing users to add, remove, and modify tasks based on priority and category. Utilizing object-oriented programming concepts, the system ensures flexibility and maintainability by incorporating encapsulation, inheritance, and polymorphism. The inclusion of a user-friendly command-line interface allows seamless interaction with the application. Additionally, the system employs efficient data structures and algorithms to ensure quick task retrieval and optimized memory usage. File handling techniques enable users to save and retrieve their tasks, ensuring data persistence across multiple sessions. This project aims to provide a practical tool that enhances productivity by simplifying task management, helping users stay organized and meet their deadlines effectively.

# II. Literature Review

The research done in this section is focused on various task management applications that have been developed or done research on. This study includes well-developed task management applications, specialized and need-based applications, and different types of existing to-do applications.

To-do list applications are need-based applications of software that create a better organized environment for the users in terms of better daily organization. Some of the most widely used to-do lists are exceptional with the execution of some of the essential and basic functions that a simple to-do list should possess. A task management application generally offers adding, removing, editing and displaying the existing tasks; however, some people demand more functionality as they need additional features from the task manager. An example of this could be ‘To-doist’ ([www.todoist.com](https://www.todoist.com)) which is a widely used app that offers many features such as group projects, project sections, auto-scheduling, due-date setup, collaboration with others, assigning tasks to others, file sharing and adding and so on.

Some task manager designs set out to develop an automation system where the user get recommendations based on the previous tasks. ‘TaskDo’ [3] designed an automation system which enables the application to collect feedback on task completion. After collecting the data over specific periods the recommender will start to make suggestions on the date, time or the type of task. For example, if the user regularly fails to complete hiking on Saturday at 11 am, and they successfully execute coffee and reading at 3 pm, then the system will recommend prioritizing coffee and reading on Saturdays. Additionally, if fishing tasks are regularly completed on Sunday morning, the system will suggest moving hiking to Sunday afternoon. Additional feedback on the suggestion will improve the quality of the system by eliminating badly selected recommendations and focusing on feedback-based evaluation.

Developing a C++ To-do list application requires clever usage of data structures such as linked lists, trees, priority queues and more. To use the right data structures there should be a predefined design for the application. If the developer wants to build a basic task manager which can add, delete and edit tasks, then they are more likely to use doubly linked lists than other data structures, because double linked list can be easily manipulated [2]. However, a more complex task manager can be developed using priority (priority-based task completion) or map and unordered map (for quick search and categorization)

In addition, there are some other approaches to the development of smart task managers that give the users better experience in terms of encouragement and reinforcement. Tamu To-Do [4] is an application developed with a gamification feature. This feature has drawn influences from some of the most popular apps like Duolingo ([www.duolingo.com](http://www.duolingo.com/)) and Habitica ([www.habitica.com](https://www.habitica.com)) which artfully use gamification to boost user activity and encourage users to be more active. For example, Duolingo offers awards and points for certain task completions such as protecting daily streak and getting exercises done. So, Tamu To-Do uses a grade-based gamification system on a scale of 0 to 100. Tamu the pet displays four different expressions 0-25 ‘very sad’, 25-50 ‘sad’ 50-75 ‘happy’ and 75-100 ‘very happy’. This is supposed to reinforce a better user experience; according to the Tamu To-Do research which got positive results, where the vast majority reported significant boost in their daily task completion in addition to emotional attachment to the pet which also boosted the productivity of the users.

The literature review investigated various existing to-do list applications and analyzed their architecture and design with the purpose of understanding common types of task management tools that are used widely and successfully. From a reward-punishment system to a feedback-based recommendation system it can be noted that there are many well-structured approaches to the development of well-functioning applications that guarantee user satisfaction and longevity.

# III. Methodology

The project was developed using Qt Project framework and Microsoft Visual Studio as the integrated development environment (IDE). Qt's cross-platform capabilities facilitated seamless development and deployment across different operating systems. Visual Studio provided robust debugging tools and enhanced code management features, essential for collaborative software development.

The application follows a Model-View-Controller (MVC) architecture to ensure separation of concerns and maintainability. The Model layer manages data persistence and business logic, while the View layer provides a user-friendly interface using Qt Widgets. The Controller layer orchestrates interactions between the Model and View, ensuring smooth data flow and application behavior.

Key functionalities include task creation, deletion, and modification, prioritization through categories or tags, and deadline management. User interactions are handled through intuitive GUI elements developed using Qt Widgets and integrated seamlessly with backend functionalities.

# IV. Conclusion

In conclusion, the development of this to-do list application has yielded significant insights and achievements for us as Computer Science students. We successfully implemented a robust software architecture using Qt Project and Visual Studio, emphasizing modular design and separation of concerns. Through iterative development and thorough testing, we ensured functionality and reliability across multiple platforms. This project not only enhanced our proficiency in software development methodologies but also improved our collaborative and problem-solving skills. Future enhancements could explore integration with cloud services for synchronization and mobile platforms for enhanced accessibility. Overall, this project underscores our growth as aspiring software engineers, equipped with the tools and knowledge to tackle real-world challenges in application development.

V. References

[1] J. Han, H. Kim, and S. Park, “TaskDo: A daily task recommender system,” in Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp), Kyoto, Japan, 2019, pp. 1–8. DOI: 10.1109/BIGCOMP.2019.8862073

[2] P. J. Bayliss and H. L. Ellis, “Right light at the right time: Development of innovative to-do list mobile application to optimize circadian lighting,” in Proc. IEEE Conf. Consumer Commun.

Netw(CCNC), Las Vegas, NV, USA, 2020, pp. 1–6. DOI: 10.1109/CCNC46108.2020.9258041

[3] A. Kumar, R. Jain, and P. Sharma, “Gamification-based to-do list mobile application development,” in Proc. IEEE Int. Conf. Recent Adv. Electron. Comput. Sci. Eng. (RAECS), Chandigarh, India, 2023, pp. 1–5. DOI: 10.1109/RAECS2023.10277897

[4] K. Nishida and K. Matsuo, “Where can we accomplish our to-do?: Estimating the target location of to-do tasks,” in Proc. IEEE Int. Conf. Ubiquitous Intell. Comput. (UIC), Philadelphia, PA, USA, 2018, pp. 55–62. DOI: 10.1109/UIC-ATC.2018.8432276 [5] S. Smith and T. Brown, “AI-assisted task management: Enhancing productivity through intelligent to-do lists,” in Proc. IEEE Int. Conf. Comput. Sci. Inf. Technol. (ICCSIT), Singapore, 2021, pp. 112–118. DOI: 10.1109/ICCSIT.2021.9633256